

Article

Polynomial-Time Constrained Message Passing for Exact MAP Inference on Discrete Models with Global Dependencies

Alexander Bauer ^{1,2,*}, Shinichi Nakajima ^{1,3,4}  and Klaus-Robert Müller ^{1,3,5,6,*}¹ Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany² BASLEARN— TU Berlin/BASF Joint Lab for Machine Learning, Technische Universität Berlin, 10587 Berlin, Germany³ Berlin Institute for the Foundations of Learning and Data, 10587 Berlin, Germany⁴ RIKEN Center for AIP, Tokyo 103-0027, Japan⁵ Department of Artificial Intelligence, Korea University, Anam-dong, Seongbuk-gu, Seoul 02841, Republic of Korea⁶ Max-Planck-Institut für Informatik, Saarland Informatics Campus E1 4, 66123 Saarbrücken, Germany

* Correspondence: alexander.bauer@tu-berlin.de (A.B.); klaus-robot.mueller@tu-berlin.de (K.-R.M.)

Abstract: Considering the worst-case scenario, the junction-tree algorithm remains the most general solution for exact MAP inference with polynomial run-time guarantees. Unfortunately, its main tractability assumption requires the treewidth of a corresponding MRF to be bounded, strongly limiting the range of admissible applications. In fact, many practical problems in the area of structured prediction require modeling global dependencies by either directly introducing global factors or enforcing global constraints on the prediction variables. However, this always results in a fully-connected graph, making exact inferences by means of this algorithm intractable. Previous works focusing on the problem of loss-augmented inference have demonstrated how efficient inference can be performed on models with specific global factors representing non-decomposable loss functions within the training regime of SSVMs. Making the observation that the same fundamental idea can be applied to solve a broader class of computational problems, in this paper, we adjust the framework for an efficient exact inference to allow much finer interactions between the energy of the core model and the sufficient statistics of the global terms. As a result, we greatly increase the range of admissible applications and strongly improve upon the theoretical guarantees of computational efficiency. We illustrate the applicability of our method in several use cases, including one that is not covered by the previous problem formulation. Furthermore, we propose a new graph transformation technique via node cloning, which ensures a polynomial run-time for solving our target problem. In particular, the overall computational complexity of our constrained message-passing algorithm depends only on form-independent quantities such as the treewidth of a corresponding graph (without global connections) and image size of the sufficient statistics of the global terms.

Keywords: algorithm; optimization; dynamic programming**MSC:** 68T99

Citation: Bauer, A.; Nakajima, S.; Müller, K.-R. Polynomial-Time Constrained Message Passing for Exact MAP Inference on Discrete Models with Global Dependencies. *Mathematics* **2023**, *11*, 2628. <https://doi.org/10.3390/math11122628>

Academic Editor: Abdullah N.

Arslan

Received: 7 April 2023

Revised: 1 June 2023

Accepted: 6 June 2023

Published: 8 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Many practical tasks can be effectively formulated as discrete optimization problems within the framework of graphical models such as Markov Random Fields (MRFs) [1–3] by representing the constraints and objective function in a factorized form. Finding the corresponding solution refers to the task of maximum a posteriori (MAP) inference, which is known to be NP-hard in general. Although there are plenty of existing approximation algorithms [4–23], several problems (described below) require finding an optimal solution. Existing exact algorithms [24–34], on the other hand, either make specific assumptions about the energy function or do not provide polynomial run-time guarantees for the worst

case. Assuming the worst-case scenario, the junction (or clique)-tree algorithm [1,35], therefore, remains the most efficient and general solution for exact MAP inference. Unfortunately, its main tractability assumption requires the treewidth [36,37] of a corresponding MRF to be bounded, strongly limiting the range of admissible applications by excluding models with global interactions. Many problems in the area of structured prediction, however, require modeling of global dependencies by either directly introducing global factors or enforcing global constraints on the prediction variables. Among the most popular use cases are (a) learning using non-decomposable (or high-order) loss functions and training via slack scaling formulation within the framework of a structural support vector machine (SSVM) [38–47], (b) evaluating generalization bounds in structured prediction [14,15,44,48–59], and (c) performing MAP inference on otherwise tractable models subject to global constraints [19,60,61]. The latter covers various search problems, including the special task of (*diverse*) *k*-best MAP inference [62,63]. Learning with non-decomposable loss functions, in particular, benefits from finding an optimal solution, as all of the theoretical guarantees of training with SSVMs assume exact inference during optimization [39,64–68].

Previous works [45–47,69] focusing on the problem of loss-augmented inference (use case (a)) have demonstrated how efficient computation can be performed on models with *specific* global factors by leveraging a dynamic programming approach based on constrained message passing. The proposed idea models non-decomposable functions as a kind of multivariate cardinality potential $\mathbf{y} \mapsto \eta(\mathbf{G}(\mathbf{y}))$, where $\eta: \mathbb{R}^P \rightarrow \mathbb{R}_+$ is some function and \mathbf{G} denotes the sufficient statistics of the global term. Although able to model popular performance measures, the objective of a corresponding inference problem is rather restricted to simple interactions between the energy of the core model F and the sufficient statistics \mathbf{G} according to $F \odot \eta(\mathbf{G})$, where $\odot: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is either a summation or a multiplication operation. Although the same framework can be applied to use case (c) by modeling global constraints via an indicator function, it cannot handle a range of other problems in use case (b) that introduce more subtle dependencies between F and \mathbf{G} .

In this paper, we extend the framework for an efficient exact inference proposed in [69] by allowing much finer interactions between the energy of the core model and the sufficient statistics of the global terms. The extended framework covers all the previous cases and applies to new problems, including the evaluation of generalization bounds in structured learning, which cannot be handled by the previous approach. At the same time, the generalization comes with no additional costs, preserving all the run-time guarantees. In fact, the resulting performance is identical to that of the previous formulation, as the corresponding modifications do not change the computational core idea of the previously proposed message passing constrained via auxiliary variables but only affect the final evaluation step (line 8 in Algorithm 1) of the resulting inference algorithm after all the required statistics have been computed. We accordingly adjust the formal statements given in [69] to ensure the correctness of the algorithmic procedure for the extended case.

Furthermore, we propose an additional graph transformation technique via node cloning that greatly improves the theoretical guarantees on the asymptotic upper bound for the computational complexity. In particular, the above-mentioned work only guarantees polynomial run-time in the case where the core model can be represented by a tree-shaped factor graph, excluding problems with cyclic dependencies. The corresponding complexity estimation for clique trees (see Theorem 2 in [69]), however, requires the maximal node degree ν to be bounded by a graph-independent constant; otherwise, it results in a term that depends exponentially on the graph size. Here, we first provide an intuition that ν tends to take on small values (see Proposition 2) and then present an additional graph transformation that reduces this parameter to a constant $\nu = 3$ (see Corollary 1). Furthermore, we analyze how the maximal number of states of auxiliary variables R , which greatly affects the resulting run-time, grows relative to the graph size (see Theorem 2).

The rest of the paper is organized as follows. In Section 2, we formally introduce the class of problems we tackle in this paper, and in Section 3, we present the constrained message-passing algorithm for finding a corresponding optimal solution based on a general representation with clique trees. Additionally, we propose a graph transformation technique via node cloning, which ensures an overall polynomial running time for the computational procedure. In Section 4, we demonstrate the expressivity of our problem formulation on several examples. For an important use case of loss-augmented inference with SSVMs, we show in Section 5 how to represent different dissimilarity measures as global cardinality potentials to align with our problem formulation. In order to validate the guarantees for the computational complexity of Algorithm 1, we present the experimental run-time measurements in Section 6. In Section 7, we provide a summary of our contributions, emphasizing the differences between our results and those in [69]. In Section 8, we broadly discuss the previous works, which is followed by the conclusions in Section 9.

2. Problem Setting

Given an MRF [1,70] over a set of discrete variables, the goal of the maximum a posteriori (MAP) problem is to find a joint variable assignment with the highest probability. This problem is equivalent to minimizing the energy of the model, which describes the corresponding (unnormalized) probability distribution over the variables. In the context of structured prediction, it is equivalent to maximizing a score or compatibility function. To avoid ambiguity, we now refer to the MAP problem as the maximization of an objective function $F : \mathbb{R}^M \rightarrow \mathbb{R}$ defined over a set of discrete variables $\mathbf{y} = (y_1, \dots, y_M)$. More precisely, we associate each function F with an MRF, where each variable y_m represents a node in the corresponding graph. Furthermore, we assume, without loss of generality, that the function F factorizes over maximal cliques \mathbf{y}_{C_t} , $C_t \subseteq \{1, \dots, M\}$ of the corresponding MRF according to

$$F(\mathbf{y}) = \sum_{t=1}^T f_t(\mathbf{y}_{C_t}). \quad (1)$$

We now use the concept of the treewidth of a graph [36] to define the complexity of a corresponding function with respect to the MAP inference, as follows.

Definition 1 (τ -decomposability). *We say that a function $F : \mathcal{D} \subseteq \mathbb{R}^M \rightarrow \mathbb{R}$ is τ -decomposable if the (unnormalized) probability $\exp(F(\mathbf{y}))$ factorizes over an MRF with a bounded treewidth τ .*

Informally, the treewidth describes the tree-likeness of a graph, that is, how well the graph structure resembles the form of a tree. In an MRF with no cycles going over the individual cliques, the treewidth is equal to the maximal size of a clique minus 1, that is, $\tau = \max_t |C_t| - 1$. Furthermore, the treewidth of a graph is considered *bounded* if it does not depend on the size of the graph in the following sense. If it is possible to increase the graph size by replicating the individual parts, the treewidth should not be affected by the number of variables in the resulting graph. One simple example is a Markov chain. Increasing the length of the chain does not affect the treewidth, which remains equal to the Markov order of that chain.

The treewidth is defined as the minimum width of a graph and can be computed algorithmically after transforming the corresponding MRF into a data structure called a junction tree or clique tree. Although the problem of constructing a clique tree with a minimum width is NP-hard in general, there are several efficient techniques [1] that can achieve good results with a width close to the treewidth.

In the following, let M be the total number of nodes in an MRF over the variables $\{y_m\}_{m=1}^M$, and let N be the maximum number of possible values each variable y_m can take on. Assuming that the maximization step dominates the time for creating a clique tree, we obtain the following known result [71]:

Proposition 1. *The computational time complexity for maximizing a τ -decomposable function is upper bounded by $O(M \cdot N^{\tau+1})$.*

The notion of τ -decomposability for real-valued functions naturally extends to mappings with multivariate outputs for which we now define joint decomposability.

Definition 2 (Joint τ -decomposability). *We say two mappings $G : \mathcal{D} \subseteq \mathbb{R}^M \rightarrow \mathbb{R}^P$ and $G' : \mathcal{D} \subseteq \mathbb{R}^M \rightarrow \mathbb{R}^{P'}$ are jointly τ -decomposable if they factorize over a common MRF with a bounded treewidth τ .*

Definition 2 ensures the existence of a common clique tree with nodes $\{C_t\}_{t=1}^T$ and the corresponding potentials $\{g_t, g'_t\}_{t=1}^T$, where $\max_t |C_t| - 1 = \tau$, and

$$G(\mathbf{y}) = \sum_{t=1}^T g_t(\mathbf{y}_{C_t}), \quad G'(\mathbf{y}) = \sum_{t=1}^T g'_t(\mathbf{y}_{C_t}).$$

Note that the individual factor functions are allowed to have fewer variables in their scope than in a corresponding clique, that is, $\text{scope}(g_t), \text{scope}(g'_t) \subseteq C_t$.

Building on the above definitions we now formally introduce a class of problem instances of MAP inference for which we later provide an exact message-passing algorithm.

Problem 1. *For $F: \mathcal{Y} \rightarrow \mathbb{R}$, $G: \mathcal{Y} \rightarrow \mathbb{R}^P$, $H: \mathbb{R} \times \mathbb{R}^P \rightarrow \mathbb{R}$ with $\mathcal{Y} \subset \mathbb{R}^M$, $|\mathcal{Y}| \leq N^M$ and $P, M, N \in \mathbb{N}_+$, we consider the following discrete optimization problem:*

$$\underset{\mathbf{y} \in \mathcal{Y}}{\text{maximize}} \quad H(F(\mathbf{y}), G(\mathbf{y})) \tag{2}$$

where we assume that (1) F and G are jointly τ -decomposable, and (2) H is non-decreasing in the first argument.

In the next section, we present multiple examples of practical problems that align with the above-mentioned abstract formulation. As our working example, here, we consider the problem of loss-augmented inference within the framework of SSVMs. This framework includes two different formulations known as margin and slack scaling, both of which require solving a combinatorial optimization problem during training. This optimization problem is either used to compute the subgradient of a corresponding objective function or find the configuration of prediction variables that violates the problem constraints the most. For example, in the case of slack scaling formulation, we can define $H(F(\mathbf{y}), G(\mathbf{y})) = F(\mathbf{y}) \cdot \eta(G(\mathbf{y}))$ for some $\eta: \mathbb{R}^P \rightarrow \mathbb{R}_+$, where $F(\mathbf{y}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$ corresponds to the compatibility score given by the inner product between a joint feature map $\Psi(\mathbf{x}, \mathbf{y})$ and a vector of trainable weights \mathbf{w} (see [39] for more details), and $\eta(G(\mathbf{y})) = \Delta(\mathbf{y}^*, \mathbf{y})$ describes the corresponding loss function for a prediction \mathbf{y} and a ground-truth output \mathbf{y}^* . In fact, a considerable number of popular loss functions used in structured prediction can generally be represented in this form, that is, as a multivariate cardinality-based potential that depends on counts of different label statistics.

3. Exact Inference for Problem 1

In this section, we derive a polynomial-time message-passing algorithm that always finds an optimal solution for Problem 1. The corresponding results can be seen as a direct extension of the well-known junction-tree algorithm.

3.1. Algorithmic Core Idea for a Simple Chain Graph

We begin by providing an intuition of why efficient inference is possible for Problem 1 using our working example of loss-augmented inference for SSVMs. For margin scaling, in the case of a linear η , the objective $F(\mathbf{y}) + \eta(G(\mathbf{y}))$ inherits the τ -decomposability directly from F and G and, therefore, can be efficiently maximized according to Proposition 1.

The main source of difficulty for slack scaling lies in the multiplication operation between $F(\mathbf{y})$ and $\eta(\mathbf{G}(\mathbf{y}))$, which results in a fully-connected MRF regardless of the form of the function η . Moreover, many popular loss functions used in structured learning require η to be non-linear, preventing efficient inference even for the margin scaling. Nevertheless, efficient inference is possible for a considerable number of practical cases, as shown below. Specifically, the global interactions between a jointly decomposable F and G can be controlled by using auxiliary variables at a polynomial cost. We now illustrate this with a simple example.

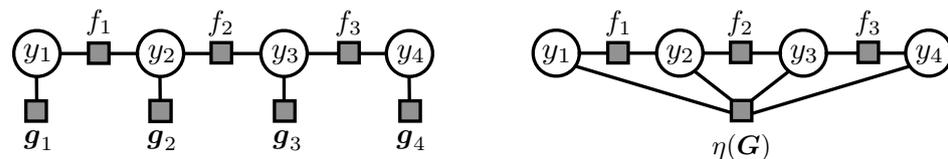


Figure 1. Factor graph representation for the margin-scaling objective, with a decomposable loss G (on the left), and a (non-decomposable) high-order loss $\eta(G)$ (on the right).

Consider a (Markov) chain of nodes with a 1-decomposable F and 0-decomposable G (e.g., Hamming distance), that is,

$$F(\mathbf{y}) = \sum_{m=1}^{M-1} f_m(y_m, y_{m+1}) \text{ and } G(\mathbf{y}) = \sum_{m=1}^M g_m(y_m). \tag{3}$$

We aim at maximizing an objective $F(\mathbf{y}) \odot \eta(\mathbf{G}(\mathbf{y}))$, where $\odot: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is a placeholder for either a summation or a multiplication operation.

The case for margin scaling with a decomposable loss $\eta(\mathbf{G}) = G$ is illustrated by the leftmost factor graph in Figure 1. Here, the corresponding factors f_m and g_m can be folded together, enabling an efficient inference according to Proposition 1. The non-linearity of η , however, can result (in the worst case!) in a global dependency between all the variable nodes, leading to a high-order potential $\eta(\mathbf{G})$, as illustrated by the rightmost factor graph in Figure 1. In slack scaling, even for a linear η , after multiplying the individual factors, we can see that the resulting model has an edge for every pair of variables, resulting in a fully-connected graph. Thus, for the last two cases, exact inference using the junction-tree algorithm is generally not feasible. The key idea of our approach is to relax the dense connections in these graphs by introducing *auxiliary variables* $\mathbf{L} = (l_1, \dots, l_M) \in \mathbb{R}^{P \times M}$ subject to the constraints

$$l_m = \sum_{k=1}^m g_k(y_k), \quad m \in \{1, \dots, M\}. \tag{4}$$

More precisely, for $H(F(\mathbf{y}), G(\mathbf{y})) = F(\mathbf{y}) \odot \eta(\mathbf{G}(\mathbf{y}))$, Problem 1 is equivalent to the following constrained optimization problem in the sense that both have the same optimal value and the same set of optimal solutions with respect to \mathbf{y} :

$$\begin{aligned} & \underset{\mathbf{y}, \mathbf{L}}{\text{maximize}} && F(\mathbf{y}) \odot \eta(\mathbf{L}_M) \\ & \text{subject to} && l_{m+1} = l_m + g_{m+1}(y_{m+1}) \quad \forall m \in \{1, \dots, M-1\} \\ & && l_1 = g_1(y_1) \end{aligned} \tag{5}$$

where the new objective involves no global dependencies and is 1-decomposable if we regard $\eta(\mathbf{L}_M)$ as a constant. We can make the local dependency structure of the new formulation more explicit by taking the constraints directly into the objective as follows:

$$Q(\mathbf{y}, \mathbf{L}) = F(\mathbf{y}) \odot \eta(\mathbf{L}_M) - \mathbb{1}_\infty[l_1 \neq g_1(y_1)] - \sum_{m=1}^{M-1} \mathbb{1}_\infty[l_{m+1} \neq l_m + g_{m+1}(y_{m+1})]. \tag{6}$$

Here, $\mathbb{1}_\alpha[\cdot]$ denotes the indicator function such that $\mathbb{1}_\alpha[\cdot] = \alpha$ if the argument in $[\cdot]$ is true and $\mathbb{1}_\alpha[\cdot] = 0$ otherwise. The indicator functions rule out the configurations that do not satisfy Equation (4) when maximization is performed. A corresponding factor graph for margin scaling is illustrated by the leftmost graph in Figure 2. We can see that our new augmented objective (6) shows only local dependencies and is, in fact, 2-decomposable.

Applying the same scheme for slack scaling yields a much more sparsely connected graph (see the rightmost graph in Figure 2). This is achieved by forcing the majority of connections to go through a single node l_4 , which we call a hub node. Actually, $Q(\mathbf{y}, \mathbf{L})$ becomes 2-decomposable if we fix the value of l_4 , which then can be multiplied into the corresponding factors of F . In this way, we can effectively reduce the overall treewidth at the expense of increased polynomial computation time (compared to a chain without the global factor), provided that the maximal number R of different states of each auxiliary variable l_m is polynomially bounded in M , which represents the number of nodes in the original graph. In the context of training SSVMs, for example, the majority of the popular loss functions satisfy this condition (see Table 1 in Section 5 for an overview).

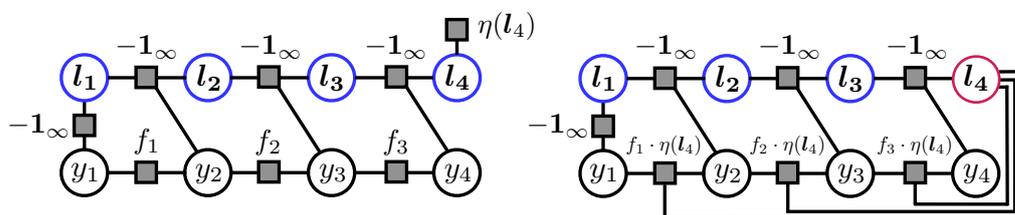


Figure 2. Factor graph representation for an augmented objective $Q(\mathbf{y}, \mathbf{L})$ for margin scaling (on the left) and for slack scaling (on the right). The auxiliary variables $\mathbf{L} = (l_1, \dots, l_4)$ are marked in blue (except l_4 for slack scaling). l_4 is the hub node.

3.2. Constrained Message-Passing Algorithm on Clique Trees

The idea presented in the previous section is intuitive and enables the reuse of existing software. Specifically, we can use the conventional junction-tree algorithm for graphical models by extending the original graph with nodes corresponding to the auxiliary variables. Alternatively, instead of performing an explicit graph transformation, we can modify the message-passing protocol, which is asymptotically at least one order of magnitude faster. Therefore, we do not explicitly introduce auxiliary variables as graph nodes before constructing the clique tree but rather use them to condition the message-passing rules. In the following, we derive an algorithm for solving an instance of Problem 1 via constrained message passing on clique trees. The resulting computational scheme can be seen as a direct extension of the junction-tree algorithm to models with specific global factors. Note that the form of tractable global dependencies is constrained according to the definition of Problem 1.

First, similar to the conventional junction-tree algorithm, we need to construct a clique tree for a given set of factors that preserves the family structure and has the running intersection property. Note that we ignore the global term during this process. The corresponding energy is given by the function F according to the definition of Problem 1. There are two equivalent approaches for constructing the clique tree [1,70]. The first is based on variable elimination and the second is based on graph triangulation, with an upper bound of $O(M \cdot N^{\tau+1})$. Figure 3 illustrates the intermediate steps of the corresponding construction process for a given factor graph using the triangulation approach. The resulting clique tree example defines the starting point for our message-passing algorithm.

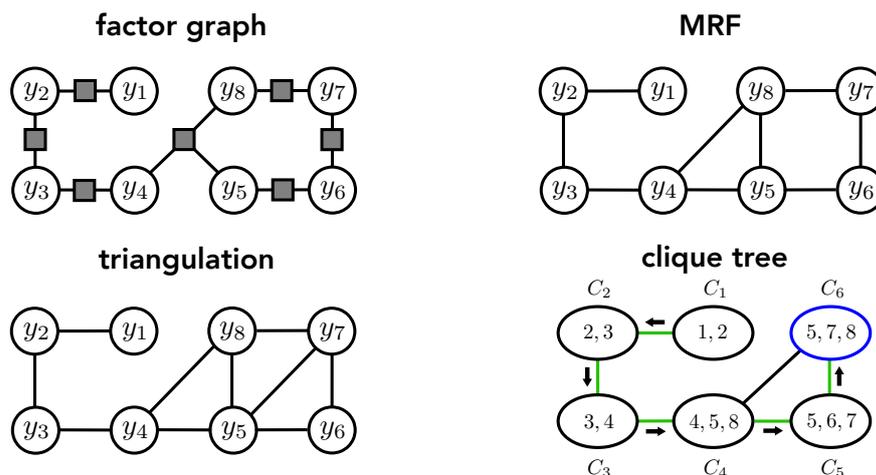


Figure 3. Illustration of the transformation process of a factor graph into a clique tree. The upper-left graph represents the original factor graph, which describes how the energy of the model without the global factor (see F in the definition of Problem 1) factorizes over the individual variables. In the first step, the factor graph is transformed into an MRF, as shown in the upper-right graph. The lower-left graph shows the result of triangulating the MRF from the previous step. Finally, the lower-right graph shows the resulting cluster graph with cliques C_1, \dots, C_6 constructed from the triangulated MRF. The numbers within the cluster nodes denote the variable indices belonging to the corresponding clique (e.g., C_5 refers to $\{y_5, y_6, y_7\}$). The green edges indicate a valid spanning clique tree extracted from the cluster graph. The dark arrows represent one possible order of message passing if clique C_6 (marked blue) has been chosen as the root of the clique tree.

Assume that a clique tree with cliques C_1, \dots, C_K is given, where C_i denotes a set of indices of variables contained in the i -th clique. We denote a corresponding set of variables by y_{C_i} . Furthermore, we use the notations $\{f_{C_i}\}_{i=1}^K$ and $\{g_{C_i}\}_{i=1}^K$ to denote the clique potentials (or factors) related to the mappings F and G in the definition of Problem 1, respectively. Additionally, we denote by C_r a clique chosen to be the root of the clique tree. Finally, we use the notation $ne(C_i)$ for the indices of the neighbors of the clique C_i . We can now compute the optimal value of the objective in Problem 1 as follows. Starting at the leaves of the clique tree, we iteratively send messages toward the root according to the following message-passing protocol. A clique C_i can send a message to its parent clique C_j if it received all messages from the rest of its neighbors C_k for $k \in ne(C_i) \setminus \{j\}$. In that case, we say that C_i is ready.

For each configuration of the variables $y_{C_i \cap C_j}$ and parameters $l_i \in \mathbb{R}^P$ (encoding the state of an auxiliary variable associated with the current clique C_i), a corresponding message from a clique C_i to a clique C_j can be computed according to the following equation:

$$\mu_{C_i \rightarrow C_j}^{l_i}(y_{C_i \cap C_j}) = \max_{y_{C_i \setminus C_j}, \{l_k\}} f_{C_i}(y_{C_i}) + \sum_{k \in ne(C_i) \setminus \{j\}} \mu_{C_k \rightarrow C_i}^{l_k}(y_{C_k \cap C_i}) \tag{7}$$

where we maximize over all configurations of the variables $y_{C_i \setminus C_j}$ and over all parameters $\{l_k\} = \{l_k : k \in ne(C_i) \setminus \{j\}\}$ subject to the following constraint

$$\sum_{k \in ne(C_i) \setminus \{j\}} l_k = l_i - g_{C_i}(y_{C_i}). \tag{8}$$

This means that each clique C_i is assigned exactly one (multivariate) auxiliary variable l_i and the range of possible values l_i can take on is implicitly defined by Equation (8). After resolving the recursion in the above equation, we can see that the variable l_i corresponds to a sum of the potentials $g_{C_k}(y_{C_k})$ for each previously processed clique C_k in a subtree of

the graph of which C_i forms the root. We refer to Equation (4) in the previous subsection for comparison.

Algorithm 1 Constrained Message Passing on a Clique Tree

Require: clique tree $\{C_i\}_i$; **Output:** optimal assignment \mathbf{y}^*

- 1: **while** root clique C_r did not receive all messages **do**
 - 2: **if** a clique C_i is ready **then**
 - 3: **for** all $\mathbf{y}_{C_i \cap C_j}$ and l_i **do**
 - 4: send a message $\mu_{C_i \rightarrow C_j}^{l_i}(\mathbf{y}_{C_i \cap C_j})$ to a parent clique C_j according to Equation (7); save the maximizing arguments $\lambda_{C_i \rightarrow C_j}^{l_i}(\mathbf{y}_{C_i \cap C_j}) := [\mathbf{y}_{C_i \setminus C_j}^*; \{l_k\}^*]$
 - 5: **end for**
 - 6: **end if**
 - 7: **end while**
 - 8: $\mathbf{l}^* \leftarrow \operatorname{argmax}_{\mathbf{l}} H(\mu(\mathbf{l}), \mathbf{l})$, where $\mu(\mathbf{l})$ is defined by Equation (9)
 - 9: Let $\mathbf{y}_{C_r}^*$ be a maximizing argument for $\mu(\mathbf{l}^*)$ in Equation (9); starting with values \mathbf{l}^* and $\mathbf{y}_{C_r}^*$, recursively reconstruct an optimal configuration \mathbf{y}^* from λ according to Equation (7).
-

The algorithm terminates if the designated root clique C_r received all messages from its neighbors. We then compute the values

$$\mu(\mathbf{l}) = \max_{\mathbf{y}_{C_r}, \{l_k\}} f_{C_r}(\mathbf{y}_{C_r}) + \sum_{k \in ne(C_r)} \mu_{C_k \rightarrow C_r}^{l_k}(\mathbf{y}_{C_k \cap C_r}) \tag{9}$$

maximizing over all configurations of \mathbf{y}_{C_r} , and $\{l_k\} = \{l_k : k \in ne(C_r)\}$ subject to the constraint $\sum_k l_k = \mathbf{l} - \mathbf{g}_{C_r}(\mathbf{y}_{C_r})$, which we use to obtain the optimal value p^* of Problem 1 according to

$$p^* = \max_{\mathbf{l}} H(\mu(\mathbf{l}), \mathbf{l}). \tag{10}$$

The corresponding optimal solution of Problem 1 can be obtained by backtracking the additional variables λ , saving optimal decisions in intermediate steps. The complete algorithm is summarized in Algorithm 1. As an alternative, we provide an additional flowchart diagram illustrating the algorithmic steps in Appendix B (see Figure A1 for further details). We underline this important result by the following theorem, for which we provide the proof in Appendix A. It should be noted that this theorem refers to the more general target objective defined in Problem 1 and should replace the corresponding statement in Theorem 2 in [69].

Theorem 1. Algorithm 1 always finds an optimal solution to Problem 1. The computational complexity is of the order $O(M \cdot N^{\tau+1} \cdot R^{v-1})$, where R denotes an upper bound on the number of states of each auxiliary variable, and v is defined as the maximal number of neighbors of a node in a corresponding clique tree.

Besides the treewidth τ , the value of the parameter v also appears to be crucial for the resulting running time of Algorithm 1 since the corresponding complexity is also exponential in v . The following proposition suggests that among all possible cluster graphs for a given MRF, there always exists a clique tree for which v tends to take on small values (provided τ is small) and effectively does not depend on the size of the corresponding MRF. We provide the proof in Appendix C.

Proposition 2. For any MRF with treewidth τ , there is a clique tree for which the maximal number of neighbors of each node is upper bounded according to $v \leq 2^{\tau+2} - 4$.

To support the above proposition, we consider the following extreme example, which is illustrated in Figure 4. We are given an MRF with a star-like shape (on the left) with $M = 7$ variables and treewidth $\tau = 1$. One valid clique tree for this MRF is shown in the middle.

In particular, the clique containing the variables y_1, y_2 has $\nu = M - 1$ neighbors. Therefore, running Algorithm 1 on that clique tree results in a computational time exponential in the graph size M . However, it is easy to modify that clique tree to have a small number of neighbors for each node (shown on the right), upper bounded by $\nu = \tau + 1 = 2$.

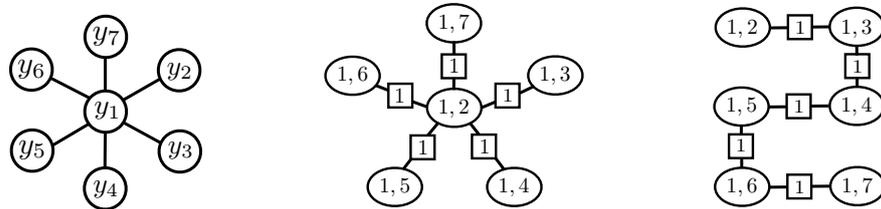


Figure 4. Illustration of an extreme example where ν can be linear in the graph size. The leftmost graph represents an MRF with $M = 7$ variables and treewidth $\tau = 1$. The graph in the middle shows a valid clique tree for the MRF on the left, where the clique $\{y_1, y_2\}$ has $M - 1$ neighbors, that is, ν is linear in the graph size for that clique tree. The rightmost graph represents another clique tree that has a chain form, where $\nu = \tau + 1 = 2$. The squared nodes denote the corresponding sepsets.

Although Proposition 2 assures the existence of a clique tree with a small ν , the actual upper bound on ν is still very pessimistic (exponential in the treewidth). In fact, by allowing a simple graph modification, we can always reduce the ν -parameter to a small constant ($\nu = 3$). Specifically, we can clone each cluster node with more than three neighbors multiple times so that each clone only carries one of the original neighbors. We then connect the clones by a chain that preserves the running intersection property. To ensure that the new cluster graph describes the same set of potentials we set the potentials for each copy of a cluster node C_i to zero: $f_{C_i}(y_{C_i}) = 0$ and $g_{C_i}(y_{C_i}) = 0$. The complete modification procedure is illustrated in Figure 5. We summarize this result in the following corollary.

Corollary 1. *Provided a given clique tree is modified according to the presented procedure for reducing the number of neighbors for each cluster node, the overall computational complexity of running Algorithm 1 (including time for graph modification) is of the order $O(M \cdot N^{\tau+1} \cdot R^2)$.*

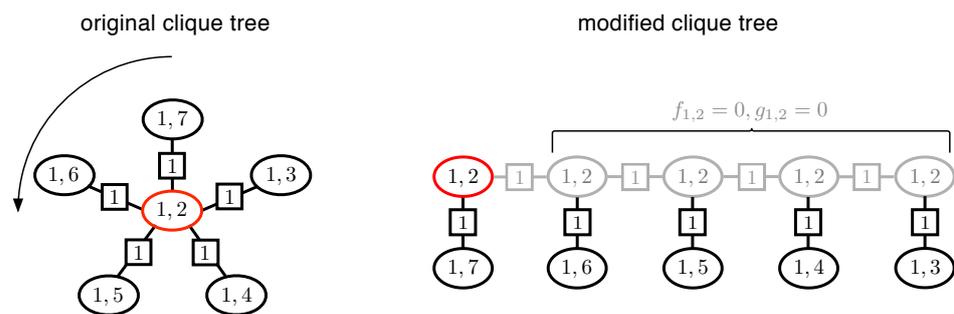


Figure 5. Illustration of a modification procedure to reduce the maximal number of neighbors ν for each cluster node in a given clique tree. The graph on the left represents an original clique tree. The only node with more than three neighbors is marked in red. We clone this cluster node multiple times so that each clone only carries one of the original neighbors. The clones are connected by a chain that preserves the running intersection property. The arrow in the left graph indicates the (arbitrarily chosen) order of processing the neighbor nodes. The graph resulting from this transformation is shown on the right. The clones in the new graph are marked in gray. To ensure that the new cluster graph describes the same set of potentials, we set the potentials for the copies of each cloned cluster node C_i to zero: $f_{C_i}(y_{C_i}) = 0$ and $g_{C_i}(y_{C_i}) = 0$. This procedure reduces the ν -parameter to a constant ($\nu = 3$), significantly reducing the computational cost.

Note that in the case where the corresponding clique tree is a chain, the resulting complexity reduces to $O(M \cdot N^{\tau+1} \cdot R)$. At this point, we would like to provide an alternative perspective of the computational complexity in this case (with $\tau = 1$, $R \sim M^2$), which shows the connection to the conventional junction-tree algorithm. Specifically, the constrained message-passing algorithm (Algorithm 1) can be seen as conventional message passing on a clique tree (for the mapping F in Problem 1) without auxiliary variables, but with an increased size of the state space for each variable y_i , from N to $N \cdot M$. Then, Proposition 1 guarantees an exact inference in time of the order $O(M \cdot (N \cdot M)^{\tau+1})$. The summation constraints with respect to the auxiliary variables can be ensured by extending the corresponding potential functions f_{C_i} to take on $-\infty$, forbidding inconsistent state transitions between individual variables. The same observation holds for message passing on factor graphs. To summarize, by introducing auxiliary variables, we can remove the global dependencies imposed by the mapping H in Problem 1, thereby reducing the overall treewidth. However, this comes at the cost of the label space becoming a function of the graph size (R is usually dependent on M).

We conclude our discussion by analyzing the relation between the maximal number of states (of the auxiliary variables) R and the number of variables M in the original MRF. In the worst case, R can grow exponentially with the graph size M . This happens, for example, when the values that the individual factors g_{C_k} can take on are scattered across a very large range that grows much faster relative to the graph size. For practical cases, however, we can assume that the individual factor functions g_{C_k} take values in an integer interval, which is either fixed or grows polynomially with the graph size. In that case, R is always a polynomial in M , rendering the overall complexity of Algorithm 1 a polynomial in the graph size, as we demonstrate with several examples in Section 6. We summarize this in the following theorem. The corresponding proof is given in Appendix D.

Theorem 2. Consider an instance of Problem 1 given by a clique tree with M variables. Let $T \in \mathbb{N}$ be a number that grows polynomially with M . Provided each factor g_{C_k} in a decomposition of G assumes values from a discrete set of integers $[-T, T] \cap \mathbb{Z}$, the number R grows polynomially with M according to $R \sim T \cdot M$.

4. Application Use Cases

In this section, we demonstrate the expressivity of Problem 1 by showcasing the diversity of existing (and potentially new) applications that align with our target objective. In particular, practitioners can gain insight into specific examples to verify whether a given task is an instance of Problem 1. The research conducted within the scope of this paper has been motivated by the following use cases:

- Learning with High-Order Loss Functions**
 SSVM enables building complex and accurate models for structured prediction by directly integrating the desired performance measure into the training objective. However, its applicability relies on the availability of efficient inference algorithms. In the state-of-the-art training algorithms, such as cutting planes [64,65], bundle methods [67,68], subgradient methods [18], and Frank–Wolfe optimization [66], inference is repeatedly performed either to compute a subgradient or find the most violating configuration. In the literature, the corresponding computational task is generally referred to as the *loss-augmented inference*, which is the main computational bottleneck during training.
- Enabling Training of Slack Scaling Formulation for SSVMs**
 The maximum-margin framework of SSVMs includes two loss-sensitive formulations known as *margin scaling* and *slack scaling*. Since the original paper on SSVMs [39], there has been much speculation about the differences in training using either of these two formulations. In particular, training via slack scaling has been conjectured to be more accurate and beneficial than margin scaling. Nevertheless, it has rarely been used in practice due to the lack of known efficient inference algorithms.

- **Evaluating Generalization Bounds for Structured Prediction**

The purpose of generalization bounds is to provide useful theoretical insights into the behavior and stability of a learning algorithm by upper bounding the expected loss or the risk of a prediction function. Evaluating such a bound could provide certain guarantees on how a system trained on some finite data will perform in the future on unseen examples. Unlike in standard regression or classification tasks with univariate real-valued outputs, in structured prediction, evaluating generalization bounds requires solving a combinatorial optimization problem, thereby limiting its use in practice [48].

- **Globally Constrained MAP Inference**

In many cases, evaluating a prediction function with structured outputs technically corresponds to performing MAP inference on a discrete graphical model, including Markov random fields (MRFs) [1], probabilistic context-free grammars (PCFGs) [72–74], hidden Markov models (HMMs) [75], conditional random fields (CRFs) [3], probabilistic relational models (PRMs) [76,77], and Markov logic networks (MLNs) [78]. In practice, we might want to modify the prediction function by imposing additional (global) constraints on its output. For example, we could perform a corresponding MAP inference subject to the constraints on the label counts specifying the size of the output or the distribution of the resulting labels, which is a common approach in applications such as sequence tagging and image segmentation. Alternatively, we might want to generate the best output with a score from a specific range that can provide deeper insights into the energy function of a corresponding model. Finally, we might want to restrict the set of possible outputs directly by excluding specific label configurations. The latter is closely related to the computational task known as (*diverse*) *k*-best MAP inference [62,63].

In the following, we provide technical details about how the generic tasks listed above can be addressed using our framework.

4.1. Loss-Augmented Inference with High-Order Loss Functions

As already mentioned, Problem 1 covers as a special case the task of loss-augmented inference (for margin and slack scaling) within the framework of SSVM [39,46]. In order to match the generic representation given in (2), we can define $F(\mathbf{y}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}) + \text{const}$, and $\eta(\mathbf{G}(\mathbf{y})) = \Delta(\mathbf{y}^*, \mathbf{y})$ for suitable $\mathbf{G}: \mathcal{Y} \rightarrow \mathbb{R}^P$ and $\eta: \mathbb{R}^P \rightarrow \mathbb{R}_+$. Here, $\Psi: \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^d$, $d \in \mathbb{N}$ denotes a joint feature map on an input–output pair (\mathbf{x}, \mathbf{y}) , $\mathbf{w} \in \mathbb{R}^d$ is a trainable weight vector, and $\Delta: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ is a dissimilarity measure between a prediction \mathbf{y} and a true output \mathbf{y}^* . Given this notation, our target objective can be written as follows:

$$H(F(\mathbf{y}), \mathbf{G}(\mathbf{y})) = F(\mathbf{y}) \odot \eta(\mathbf{G}(\mathbf{y})), \quad \odot \in \{+, \cdot\}. \quad (11)$$

We note that a considerable number of non-decomposable (or high-order) loss functions in structured prediction can be represented as multivariate cardinality-based potentials $\mathbf{y} \mapsto \eta(\mathbf{G}(\mathbf{y}))$, where the mapping \mathbf{G} encodes the label statistics, e.g., the number of true or false positives with respect to the ground truth. Furthermore, the maximal number of states R for the corresponding auxiliary variables related to \mathbf{G} is polynomially bounded in the number of variables M (see Table 1 in Section 5 for an overview of existing loss functions and the resulting values for R). For the specific case of a chain graph with F_β -loss, for example, the resulting complexity $O(M^3 \cdot N^2)$ of Algorithm 1 is cubic in the graph size.

4.2. Evaluating Generalization Bounds in Structured Prediction

In the following, we demonstrate how our algorithmic idea can be used to evaluate the PAC-Bayesian generalization bounds for max-margin structured prediction. As a working example, we consider the following generalization theorem, as stated in [48]:

Theorem 3. Assume that $0 \leq \Delta(\mathbf{y}^*, \mathbf{y}) \leq 1$. With a probability of at least $1 - \delta$ over the draw of the training set $\mathcal{S} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$ of size $n \in \mathbb{N}$, the following holds simultaneously for all weight vectors \mathbf{w} :

$$\mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \rho} [\Delta(\mathbf{y}, h_{\mathbf{w}}(\mathbf{x}))] \leq \frac{\|\mathbf{w}\|^2}{n} + \sqrt{\frac{\|\mathbf{w}\|^2 \ln(\frac{2dn}{\|\mathbf{w}\|^2}) + \ln(\frac{n}{\delta})}{2(n-1)}} \tag{12}$$

$$+ \frac{1}{n} \sum_{i=1}^n \max_{\hat{\mathbf{y}}} \mathbb{1}_1 \left[\mathbf{w}^\top (\Psi(\mathbf{x}_i, \mathbf{y}_i) - \Psi(\mathbf{x}_i, \hat{\mathbf{y}})) \leq \Delta_{HD}(\mathbf{y}_i, \hat{\mathbf{y}}) \right] \cdot \Delta(\mathbf{y}_i, \hat{\mathbf{y}})$$

where $h_{\mathbf{w}}(\mathbf{x}) = \underset{\hat{\mathbf{y}}}{\operatorname{argmax}} \mathbf{w}^\top \Psi(\mathbf{x}, \hat{\mathbf{y}})$ denotes the corresponding prediction function.

Evaluating the second term on the right-hand side of the inequality in (12) involves a maximization over $\mathbf{y} \in \mathcal{Y}$ for each data point $(\mathbf{x}, \mathbf{y}^*)$ according to

$$\max_{\mathbf{y} \in \mathcal{Y}} \mathbb{1}_1 \left[\mathbf{w}^\top (\Psi(\mathbf{x}, \mathbf{y}^*) - \Psi(\mathbf{x}, \mathbf{y})) \leq \Delta_{HD}(\mathbf{y}^*, \mathbf{y}) \right] \cdot \Delta(\mathbf{y}^*, \mathbf{y}).$$

We now show that this maximization term is an instance of Problem 1. More precisely, we consider an example with $\tau > 1$ and an F_1 -loss (see Table 1). Next, we define $F(\mathbf{y}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$, $\eta(G(\mathbf{y})) = \Delta_{F_1}(\mathbf{y}^*, \mathbf{y})$, and set

$$H(F(\mathbf{y}), G(\mathbf{y})) = \mathbb{1}_1 \left[\mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}^*) - F(\mathbf{y}) \leq |\mathbf{y}^*| - G_1(\mathbf{y}) + G_2(\mathbf{y}) \right] \cdot \eta(G(\mathbf{y}))$$

where we use $\Delta_{HD}(\mathbf{y}^*, \mathbf{y}) = FP + FN$, $FN = |\mathbf{y}^*| - TP$, which removes the need for additional auxiliary variables for the Hamming distance, reducing the resulting computational cost. Here, TP , FP , and FN denote the numbers of true positives, false positives, and false negatives, respectively. $|\mathbf{y}^*|$ denotes the size of the output \mathbf{y}^* . Both $|\mathbf{y}^*|$ and $\mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}^*)$ are constant with respect to the maximization over \mathbf{y} . Note also that H is non-decreasing in $F(\mathbf{y})$. Furthermore, the number of states of the auxiliary variables is upper bounded by $R = M^2$ (see Table 1). Therefore, here, the computational complexity of Algorithm 1 (according to Corollary 1) is given by $O(M^5 \cdot N^{\tau+1})$.

As a final remark, we note that training an SSVM corresponds to solving a convex problem but is not consistent. It fails to converge to the optimal predictor even in the limit of infinite training data (see [60] for more details). However, minimizing the (non-convex) generalization bound is consistent. Algorithm 1 provides an effective evaluation tool that could potentially be used for the development of new training algorithms based on the direct minimization of such bounds. We leave the corresponding investigation for future work.

4.3. Globally-Constrained MAP Inference

Another common use case is performing MAP inference on graphical models (such as MRFs) subject to additional constraints on the variables or the range of the corresponding objective including various tasks such as image segmentation in computer vision, sequence tagging in computational biology or natural language processing, and signal denoising in information theory. We note that an important tractability assumption in the definition of Problem 1 is the τ -decomposability of F and G with a reasonably small treewidth τ . In areas such as computer vision, we usually encounter models (e.g., Ising grid model) where the treewidth is a function of the graph size given by $\tau = \sqrt{M}$. In this case, we can use

Algorithm 1 by leveraging the technique of dual decomposition [5,16,79]. More precisely, we decompose the original graph in (multiple) trees, where the global factor is attached to exactly one of the trees in our decomposition. We also note that from a technical perspective, the problem of MAP inference subject to some global constraints on the statistics $G(\mathbf{y})$ is equivalent to the MAP problem augmented with a global cardinality-based potential $\eta(G(\mathbf{y}))$. Specifically, we can define η as an indicator function $\mathbb{1}_{-\infty}[\cdot]$, which returns $-\infty$ if the corresponding constraint on $G(\mathbf{y})$ is violated. Furthermore, the form of η does not affect the message passing of the presented algorithm. We can always check the validity of a corresponding constraint after all the necessary statistics have been computed.

4.3.1. Constraints on Label Counts

As a simple example, consider the binary-sequence tagging experiment, that is, every output $\mathbf{y} \in \mathcal{Y}$ is a sequence, and each site in the sequence can be either 0 or 1. Given some prior information on the number b of positive labels, we can improve the quality of the results by imposing a corresponding constraint on the outputs:

$$\begin{aligned} & \underset{\mathbf{y} \in \mathcal{Y}}{\text{maximize}} && \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}) \\ & \text{subject to} && \sum_{i=1}^M y_i = b \end{aligned} \tag{13}$$

We can write this as an instance of Problem 1 by setting

$$H(F(\mathbf{y}), G(\mathbf{y})) = F(\mathbf{y}) + \mathbb{1}_{-\infty}[G(\mathbf{y}) \neq b], \tag{14}$$

where $F(\mathbf{y}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$, and $G(\mathbf{y}) = \sum_{i=1}^M y_i$. Since all the variables in \mathbf{y} are binary, the number of states R of the corresponding auxiliary variables $l_m = \sum_{i=1}^m y_i$ is upper bounded by M . In addition, because the output graph is a sequence, we have $\tau = 1, \nu = 2$. Therefore, here, the computational complexity of Algorithm 1 is of the order $O(M^2 \cdot N^2)$.

4.3.2. Constraints on Objective Value

We continue with the binary-sequence tagging example (with pairwise interactions). To force constraints on the score to be in a specific range, as in

$$\begin{aligned} & \underset{\mathbf{y} \in \mathcal{Y}}{\text{maximize}} && \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}) \\ & \text{subject to} && a \leq \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}) \leq b \end{aligned} \tag{15}$$

we first rewrite the prediction function in terms of its sufficient statistics according to

$$\mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}) = \sum_{o,s} w_{o,s} \underbrace{\sum_{t=1}^M \mathbb{1}_1[x_t = o \wedge y_t = s]}_{=:G_{o,s}(\mathbf{y})} + \sum_{s_1,s_2} w_{s_1,s_2} \underbrace{\sum_{t=2}^M \mathbb{1}_1[y_{t-1} = s_1 \wedge y_t = s_2]}_{=:G_{s_1,s_2}(\mathbf{y})}, \tag{16}$$

$\mathbf{w} = (\dots, w_{o,s}, \dots, w_{s_1,s_2}, \dots)$, $\mathbf{G} = (\dots, G_{o,s}, \dots, G_{s_1,s_2}, \dots)$, and define

$$H(F(\mathbf{y}), \mathbf{G}(\mathbf{y})) = F(\mathbf{y}) + \mathbb{1}_{-\infty}[\mathbf{w}^\top \mathbf{G}(\mathbf{y}) \notin [a, b]]. \tag{17}$$

Note that \mathbf{G} contains all the sufficient statistics of F such that $F(\mathbf{y}) = \mathbf{w}^\top \mathbf{G}(\mathbf{y})$. Here, we could replace $a \leq \mathbf{w}^\top \Psi(\mathbf{y}) \leq b$ with any (non-linear) constraint on the sufficient statistics of the joint feature map $\Psi(\mathbf{y})$.

The corresponding computational complexity can be derived by considering an urn problem: one with $D \cdot N$ and one with N^2 distinguishable urns and M indistinguishable balls. Here, D denotes the size of the dictionary for the observations x_i in the input sequence x . Note that the dictionary of the input symbols can be large compared to other problem parameters. However, we can reduce D to the size of the vocabulary only occurring in the current input x . The first urn problem corresponds to the unary observation-state statistics $G_{o,s}(\mathbf{y})$, and the second corresponds to the pairwise statistics for the state transition $G_{s_1,s_2}(\mathbf{y})$. The resulting number of possible distributions of balls over the urns is given by

$$\underbrace{\binom{M + D \cdot N - 1}{M}}_{\leq M^{D \cdot N}} \cdot \underbrace{\binom{M + N^2 - 1}{M}}_{\leq M^{N^2}} \leq \underbrace{M^{D \cdot N + N^2}}_{=R} \tag{18}$$

Although the resulting complexity (due to $v = 2$) being $O(M^{D \cdot N + N^2 + 1} \cdot N^2)$ is still a polynomial in the number of variables M , the degree is quite high, making it suitable for only short sequences. For practical use, we recommend the efficient approximation framework of Lagrangian relaxation and Dual Decomposition [5,16,79].

4.3.3. Constraints on Search Space

The constraints on the search space can be different from the constraints we can impose on the label counts. For example, we might want to exclude a set of K complete outputs $\{\mathbf{y}^1, \dots, \mathbf{y}^K\}$ from the feasible set \mathcal{Y} by using an exclusion potential $\mathbb{1}_{-\infty}[\mathbf{y} \in \{\mathbf{y}^1, \dots, \mathbf{y}^K\}]$.

For simplicity, we again consider a sequence-tagging example with pairwise dependencies. Given a set of K patterns to exclude, we can introduce auxiliary variables $l_m \in \{0, 1\}^K$, where for each pattern \mathbf{y}^k , we have a constraint $(l_m)_k = \max\{\mathbb{1}_1[y_m^k \neq y_m], (l_{m-1})_k\}$. More precisely, we modify the message computation in (7) with respect to the auxiliary variables by replacing the corresponding constraints $(l_m)_k = (l_{m-1})_k + \mathbb{1}_1[y_m^k \neq y_m]$ in the maximization over $\{l_m\}$ with the constraints $(l_m)_k = \max\{\mathbb{1}_1[y_m^k \neq y_m], (l_{m-1})_k\}$. Therefore, the maximal number of states for l_m is given by $R = 2^K$. The resulting complexity for finding an optimal solution over $\mathbf{y} \in \mathcal{Y} \setminus \{\mathbf{y}^1, \dots, \mathbf{y}^K\}$ is of the order $O(2^K \cdot M \cdot N^2)$.

A related problem is finding a *diverse* k -best solution. Here, the goal is to produce the best solutions that are sufficiently different from each other according to a diversity function, e.g., a loss function such as the Hamming distance Δ_{HD} . More precisely, after computing the MAP solution \mathbf{y}^1 , we compute the second-best (diverse) output \mathbf{y}^2 with $\Delta_{HD}(\mathbf{y}^1, \mathbf{y}^2) \geq m_1$. For the third-best solution, we then require $\Delta_{HD}(\mathbf{y}^1, \mathbf{y}^3) \geq m_2$ and $\Delta_{HD}(\mathbf{y}^2, \mathbf{y}^3) \geq m_2$, and so on. In other words, we search for an optimal output \mathbf{y}^k such that $\Delta_{HD}(\mathbf{y}^k, \mathbf{y}^K) \geq m_{K-1}$, $m_k \in \mathbb{N}$ for all $k \in \{1, \dots, K - 1\}$.

For this purpose, we define auxiliary variables $l_m \in \{0, \dots, M\}^{K-1}$, where for each pattern \mathbf{y}^k , we have a constraint $(l_m)_k = (l_{m-1})_k + \mathbb{1}_1[y_m^k \neq y_m]$, which computes the Hamming distance of a solution \mathbf{y} with respect to the pattern \mathbf{y}^k . Therefore, we can define

$$H(F(\mathbf{y}), G(\mathbf{y})) = F(\mathbf{y}) + \mathbb{1}_{-\infty}[\exists k \in \{1, \dots, K - 1\}: G_k(\mathbf{y}) < m_k] \tag{19}$$

where $G = (G_1, \dots, G_{K-1})$, and at the final stage (due to $G(\mathbf{y}) = l_M$), we have all the necessary information to evaluate the constraints with respect to the diversity function (here Hamming distance). The maximal number of states R for the auxiliary variables is upper bounded by M^{K-1} . Therefore, the resulting running time for finding the K -th diverse output sequence is of the order $O(M^K \cdot N^{\tau+1})$.

Finally, we note that the concept of diverse K -best solutions can also be used during the training of SSVMs to speed up the convergence of a corresponding algorithm by generating diverse cutting planes or subgradients, as described in [63]. An appealing property of Algorithm 1 is that we obtain some of the necessary information for free as a side effect of the message passing.

5. Compact Representation of Loss Functions

We now further advance the task of loss-augmented inference (see Section 4.1) by presenting a list of popular dissimilarity measures that our algorithm can handle, which are summarized in Table 1. The measures are given in a compact representation $\Delta(\mathbf{y}^*, \mathbf{y}) = \eta(\mathbf{G}(\mathbf{y}))$ based on the corresponding sufficient statistics encoded as a mapping \mathbf{G} . Columns 2 and 3 show the form of $\mathbf{G}(\cdot)$ and η , respectively. Column 4 provides an upper bound R on the number of possible values of auxiliary variables that affect the resulting running time of Algorithm 1 (see Corollary 1).

Table 1. Compact representation of popular dissimilarity measures based on the corresponding sufficient statistics $\mathbf{G}(\cdot)$. The upper (and lower!) bounds on the number of states of the auxiliary variables R for the presented loss functions are shown in the last column.

Loss	$\mathbf{G}(\mathbf{y})$	$\eta(\mathbf{G}(\cdot))$	R
$\Delta_{0/1}$	$(TP, \mathbb{1}[FP > 0])$	$\mathbb{1}[\max\{M - G_1, G_2\} > 0]$	M^2
Δ_{HD}	$\sum_{t=1}^M \mathbb{1}[y_t^* \neq y_t]$	G	M
Δ_{HL}	$\sum_{t=1}^M \mathbb{1}[y_t^* \neq y_t]$	G/M	M
Δ_{WHD}	$\{\#(s_1, s_2)\}_{s_1, s_2 \in \{1, \dots, N\}}$	$\sum_{s_1, s_2} \text{weight}(s_1, s_2) \cdot G_{s_1, s_2}$	M^{N^2}
$\Delta_{\#FP}$	FP	G	M
Δ_R	TP	$1 - G/ \mathbf{y}^* $	M
Δ_P	(TP, FP)	$1 - \frac{G_1}{G_1 + G_2}$	M^2
$\Delta_{F\beta}$	(TP, FP)	$1 - \frac{(1 + \beta^2) \cdot G_1}{\beta^2 \cdot \mathbf{y}^* + G_1 + G_2}$	M^2
$\Delta_{\cap/\cup}$	(TP, FP)	$1 - \frac{G_1}{ \mathbf{y}^* + G_2}$	M^2
Δ_{LC}	$\sum_{t=1}^M y_t$	$ G - \sum_{t=1}^M y_t^* $	M
$\Delta_{\#CB}$	$\#CB$	G	M
Δ_{CBR}	$(\#CB, \mathbf{y})$	G_1/G_2	M^2
Δ_{BLEU}	$(TP_1, FP_1, \dots, TP_K, FP_K)$	$1 - BP(\cdot) \cdot \exp\left(\frac{1}{K} \sum_{k=1}^K \log p_k\right)$	M^{2K}
$\Delta_{ROUGE-K}$	$\{\text{count}(k, X)\}_{k \in \text{grams}(Ref)}$	$1 - \frac{\sum_{S \in Ref} \sum_{k \in \text{grams}(S)} \min\{\text{count}(k, S), G_k\}}{\sum_{S \in Ref} \sum_{k \in \text{grams}(S)} \text{count}(k, S)}$	M^D
$\Delta_{ROUGE-LCS}$	$\{LCS(X, S)\}_{S \in Ref}$	$1 - \frac{1}{ Ref } \sum_{S \in Ref} \frac{(1 + \beta^2)P(G_S) \cdot R(G_S)}{\beta^2 P(G_S) + R(G_S)}$	$M^{2 Ref }$

Here, $|\mathbf{y}| = M$ denotes the number of nodes in the output \mathbf{y} . TP, FP , and FN are the numbers of true positives, false positives, and false negatives, respectively. The number of true positives for a prediction \mathbf{y} and a true output \mathbf{y}^* is defined as the number of common nodes with the same label. The number of false positives is determined by the number of nodes that are present in the output \mathbf{y} but missing (or with another label) in the true output \mathbf{y}^* . Similarly, the number of false negatives corresponds to the number of nodes present in \mathbf{y}^* but missing (or with another label) in \mathbf{y} . In particular, it holds that $|\mathbf{y}^*| = TP + FN$.

We can see in Table 1 that each element of $\mathbf{G}(\cdot)$ is a sum of binary variables, which significantly reduces the image size of mapping $\mathbf{G}(\cdot)$. This occurs despite the exponential variety of the output space \mathcal{Y} . As a result, the image size grows only polynomially with the size of the outputs $\mathbf{y} \in \mathcal{Y}$, and the number R provides an upper bound on the image size of $\mathbf{G}(\cdot)$.

Zero-One Loss ($\Delta_{0/1}$)

This loss function takes on binary values $\{0, 1\}$ and is the most uninformative since it requires a prediction to match the ground truth to 100% and provides no partial quan-

tification of the prediction quality in the opposite case. Technically, this measure is not decomposable since it requires the numbers of *FP* and *FN* to be evaluated via

$$\Delta_{0/1}(\mathbf{y}^*, \mathbf{y}) = \mathbb{1}[\max\{FP, FN\} > 0]. \tag{20}$$

Sometimes, we cannot compute the *FN* (unlike *FP*) from the individual nodes of a prediction. Instead, we can count the *TP* and compute the *FN* using the relationship $|\mathbf{y}^*| = TP + FN$. For example, if the outputs $\mathbf{y} \in \mathcal{Y}$ are set with no ordering indication of the individual set elements, we need to know the whole set \mathbf{y} in order to be able to compute the *FN*. Therefore, computing *FN* from a partially constructed output is not possible. We note, however, that in the case of the zero-one loss function, there is a faster inference approach, which involves modifying the prediction algorithm to also compute the second-best output and selecting the best result based on the value of the objective function.

Hamming Distance/Hamming Loss (Δ_{HD}, Δ_{HL})

In the context of sequence learning, given a true output \mathbf{y}^* and a prediction \mathbf{y} of the same length, the *Hamming distance* measures the number of states on which the two sequences disagree:

$$\Delta_{HD}(\mathbf{y}^*, \mathbf{y}) = \sum_{t=1}^M \mathbb{1}[\mathbf{y}_t^* \neq \mathbf{y}_t]. \tag{21}$$

By normalizing this value, we obtain the *Hamming loss*, which does not depend on the length of the sequences. Both measures are decomposable.

Weighted Hamming Distance (Δ_{WHD})

For a given matrix weight $\in \mathbb{R}^{N \times N}$, the *weighted Hamming distance* is defined as $\Delta_{WHD}(\mathbf{y}^*, \mathbf{y}) = \sum_{t=1}^M \text{weight}(\mathbf{y}_t^*, \mathbf{y}_t)$. However, keeping track of the accumulated sum of the weights until the current position t in a sequence, unlike for the Hamming distance, can be intractable. We can, however, use the following observation. It is sufficient to count the occurrences $(\mathbf{y}_t^*, \mathbf{y}_t)$ for each pair of states $\mathbf{y}_t^*, \mathbf{y}_t \in \{1, \dots, N\}$ according to

$$\sum_{t=1}^M \text{weight}(\mathbf{y}_t^*, \mathbf{y}_t) = \sum_{s_1, s_2} \text{weight}(s_1, s_2) \sum_{t=1}^M \mathbb{1}[\mathbf{y}_t^* = s_1 \wedge \mathbf{y}_t = s_2]. \tag{22}$$

In other words, each dimension of \mathbf{G} (denoted as G_{s_1, s_2}) corresponds to

$$G_{s_1, s_2}(\mathbf{y}; \mathbf{y}^*) = \sum_{t=1}^M \mathbb{1}[\mathbf{y}_t^* = s_1 \wedge \mathbf{y}_t = s_2]. \tag{23}$$

Here, we can upper bound the image size of $\mathbf{G}(\cdot)$ by considering an urn problem with N^2 distinguishable urns and M indistinguishable balls. The number of possible distributions of the balls over the urns is given by $\binom{M+N^2-1}{M} \leq M^{N^2}$.

False Positives/Precision/Recall ($\Delta_{\#FP}, \Delta_P, \Delta_R$)

False positives measure the discrepancy between outputs by counting the number of false positives in a prediction \mathbf{y} with respect to the true output \mathbf{y}^* . This metric is often used in learning tasks such as natural language parsing due to its simplicity. Precision and recall are popular measures used in information retrieval. By subtracting the corresponding values from one, we can easily convert them to a loss function. Unlike precision given by $TP / (TP + FP)$, recall effectively depends on only one parameter. Although it is originally parameterized by two parameters given as $TP / (TP + FN)$, we can exploit the fact that the value $|\mathbf{y}^*| = TP + FN$ is always known in advance during the inference, rendering recall a decomposable measure.

F_β -Loss (Δ_{F_β})

The $F_{\beta=1}$ -score is often used to evaluate performance in various natural language processing applications and is also suitable for many structured prediction tasks. It is originally defined as the harmonic mean of precision and recall

$$F_1 = \frac{2TP}{2TP + FP + FN}. \tag{24}$$

However, since the value $|\mathbf{y}^*| = TP + FN$ is always known in advance during the inference, the F_β -score effectively depends on only two parameters (TP, FP). The corresponding loss function is defined as $\Delta_{F_\beta} = 1 - F_\beta$.

Intersection Over Union ($\Delta_{\cap/\cup}$)

The *Intersection-Over-Union* loss is mostly used in image processing tasks such as image segmentation and object recognition and was used as a performance measure in the Pascal Visual Object Classes Challenge [80]. It is defined as $1 - \text{area}(\mathbf{y}^* \cap \mathbf{y}) / \text{area}(\mathbf{y}^* \cup \mathbf{y})$. We can easily interpret this value in cases where the outputs \mathbf{y}^*, \mathbf{y} describe the bounding boxes of pixels. The more the overlap of two boxes, the smaller the loss value. We note that in the case of binary image segmentation, for example, we have a different interpretation of true and false positives. In particular, it holds that $TP + FN = P$, where P is the number of positive entries in \mathbf{y}^* . In terms of the contingency table, this yields

$$\Delta_{\cap/\cup} = 1 - \frac{TP}{(TP + FP + FN)}. \tag{25}$$

Since $|\mathbf{y}^*| = TP + FN$, the value $\Delta_{\cap/\cup}$ effectively depends on only two parameters (instead of three). Moreover, unlike F_β -loss, $\Delta_{\cap/\cup}$ defines a proper distance metric on sets.

Label-Count Loss (Δ_{LC})

The *Label-Count* loss is a performance measure used for the task of binary image segmentation in computer vision and is given by

$$\Delta(\mathbf{y}^*, \mathbf{y}) = \frac{1}{M} \left| \sum_{i=1}^M y_i - \sum_{i=1}^M y_i^* \right|. \tag{26}$$

This loss function prevents assigning low energy to segmentation labelings with substantially different areas compared to the ground truth.

Number/Rate of Crossing Brackets ($\Delta_{\#CB}, \Delta_{CBR}$)

The number of *Crossing Brackets* ($\#CB$) is a measure used to evaluate performance in natural language parsing. It computes the average of how many constituents in one tree \mathbf{y} cross over constituent boundaries in the other tree \mathbf{y}^* . The normalized version (by $|\mathbf{y}|$) of this measure is called the *Crossing Brackets (Recall) Rate*. Since the value $|\mathbf{y}|$ is not known in advance, the evaluation requires a further parameter for the size of \mathbf{y} .

Bilingual Evaluation Understudy (Δ_{BLEU})

Bilingual Evaluation Understudy, or *BLEU* for short [81], is a measure used to evaluate the quality of machine translations. It computes the geometric mean of the precision $p_k = TP_k / (TP_k + FP_k)$ of k -grams of various lengths (for $k = 1, \dots, K$) between a hypothesis and a set of reference translations, multiplied by a factor $BP(\cdot)$ to penalize short sentences according to

$$\Delta_{BLEU}(\mathbf{y}^*, \mathbf{y}) = 1 - BP(\mathbf{y}) \cdot \exp\left(\frac{1}{K} \sum_{k=1}^K \log p_k\right). \tag{27}$$

Note that K is a constant, rendering the term M^{2K} a polynomial in M .

Recall-Oriented Understudy for Gisting Evaluation ($\Delta_{ROUGE-K}, \Delta_{ROUGE-LCS}$)

The *Recall-Oriented Understudy for Gisting Evaluation*, or *ROUGE* for short [82], is a measure used to evaluate the quality of a summary by comparing it to other summaries created by humans. More precisely, for a given set of reference summaries *Ref* and a summary candidate *X*, ROUGE-K computes the percentage of k-grams from *Ref* that appear in *X* according to

$$ROUGE-K(X, Ref) = \frac{\sum_{S \in Ref} \sum_{k \in k\text{-grams}(S)} \min\{\text{count}(k, X), \text{count}(k, S)\}}{\sum_{S \in Ref} \sum_{k \in k\text{-grams}(S)} \text{count}(k, S)} \tag{28}$$

where $\text{count}(k, S)$ provides the number of occurrences of a k-gram *k* in a summary *S*. We can estimate an upper bound *R* on the image size of $G(\cdot)$ similarly to the derivation for the weighted Hamming distance above as M^D , where $D := |\text{grams}(Ref)|$ is the dimensionality of $G(\cdot)$. This represents the number of unique k-grams occurring in the reference summaries. Note that we do not need to count grams that do not occur in the references.

Another version, *ROUGE-LCS*, is based on the concept of the longest common subsequence (LCS). More precisely, for two summaries *X* and *Y*, we first compute $LCS(X, Y)$, which is the length of the LCS. We then use this value to define precision and recall measures given by $LCS(X, Y)/|X|$ and $LCS(X, Y)/|Y|$, respectively. These measures are used to evaluate the corresponding *F*-measure:

$$\Delta_{ROUGE-LCS} = 1 - \frac{1}{|Ref|} \sum_{S \in Ref} \frac{(1 + \beta^2)P_{LCS(X,S)} \cdot R_{LCS(X,S)}}{\beta^2 P_{LCS(X,S)} + R_{LCS(X,S)}}. \tag{29}$$

In other words, each dimension in $G(\cdot)$ is indexed by an $S \in Ref$. *ROUGE-LCS* (unlike *ROUGE-K*) is non-decomposable.

6. Validation of Theoretical Time Complexity

To demonstrate feasibility, we evaluate the performance of our algorithm on several application tasks: part-of-speech tagging [83], base-NP chunking [84], and constituency parsing [39,46]. More precisely, we consider the task of loss-augmented inference (see Section 4.1) for margin and slack scaling with different loss functions. The run-times for the task of diverse k-best MAP inference (Section 4.3) and for evaluating the structured generalization bounds (see Section 4.2) are identical to the run-time for the loss-augmented inference with slack scaling. We omit the corresponding plots due to redundancy. In all the experiments, we used Penn English Treebank-3 [85] as a benchmark data set, which provides a large corpus of annotated sentences from the Wall Street Journal. For better visualization, we restrict our experiments to sentences containing at most 40 words. The resulting time performance is shown in Figure 6. We can see that different loss functions result in different computation costs, as indicated by the number of values for the auxiliary variables shown in the last column of Table 1. In particular, the shapes of the curves are consistent with the upper bound provided in Theorem 1, which reflects the polynomial degree of the overall dependency with respect to the graph size *M*. The difference between margin and slack scaling is due to the fact that in the case of a decomposable loss function *G*, the corresponding loss terms can be folded into the factors of the compatibility function *F*, allowing for the use of conventional message passing.

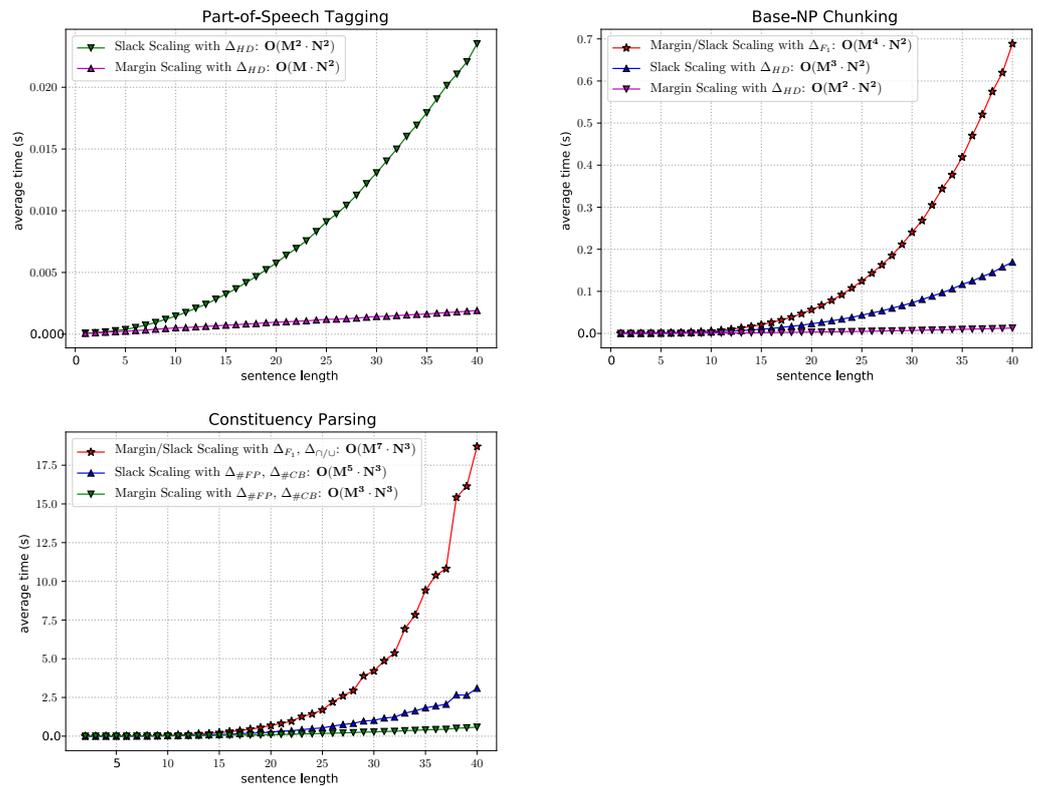


Figure 6. Empirical evaluation of the run-time performance of loss-augmented inference for part-of-speech tagging, base-NP chunking, and constituency-parsing tasks.

7. Summary of Contributions

In this paper, we provided a set of formal contributions by generalizing the idea of constrained message passing introduced in [69]. Our improved framework, which is presented in this paper, includes the following key elements:

- Abstract definition of the target problem (see Problem 1);
- Constrained message-passing algorithm on clique trees (see Algorithm 1);
- Formal statements to ensure theoretical properties such as correctness and efficiency (see Theorem 1, Proposition 2, Corollary 1, Theorem 2).

We emphasize that the idea of using auxiliary variables to constrain message passing is not novel per se and was originally proposed in [69]. Our **first contribution** concerns the difference in the definition of the target problem, which broadens the range of admissible applications. More precisely, the target objective H in the previous publication relates the energy of the model F and the vector-valued mapping G , which describes the sufficient statistics of the global terms in a rather restricted form according to

$$H(F(\cdot), G(\cdot)) := F(\cdot) \odot \eta(G(\cdot)), \quad \odot \in \{+, \cdot\}$$

where η is a non-negative function on G . Although this is sufficient for the purposes of [69], there are other computational problems that could benefit from the same message-passing idea but cannot be addressed by it in the above form. Therefore, we relax the form of the function H in our target problem (Problem 1) to allow much finer interactions between F and G . Specifically, we allow H to be **any function** on the arguments (F, G) , which is **restricted** only by the requirement that it is **non-decreasing in the first argument**. Without this assumption, the proposed algorithm (Algorithm 1) is not guaranteed to provide an optimal solution. Therefore, all related theorems, algorithms, and the corresponding proofs in [69] must be adjusted accordingly. Conceptually, we can reuse the computational core idea, including the introduction of auxiliary variables, without any changes. The only

difference in the new version of the algorithm is how the statistics gathered during message passing are evaluated at the end to ensure the optimality of the solution according to the new requirement on the function H (see line 8 in Algorithm 1). In order to motivate the significance and practical usability of this extension, we demonstrated that mappings F and G can interact by means of the function H in a highly non-trivial way, thereby exceeding the range of problem formulations in [69]. In particular, we showed how the generalization bounds in structured prediction can be evaluated using our approach (see Section 4.2). To the best of our knowledge, there are no existing methods for finding an optimal solution for this task.

Our **second contribution** involves a new graph transformation technique via node cloning (illustrated in Figure 5), which significantly enhances the asymptotic bounds on the computational complexity of Algorithm 1. Previously, in [69], the polynomial running time was guaranteed only if the maximal node degree ν in a corresponding cluster graph was bounded by a (small) constant. The proposed transformation effectively removes this limitation and ensures **polynomial running time** regardless of the graph structure. We emphasize this result in Corollary 1. Note that the term $R^{\nu-1}$ in Theorem 1 in [69] has been replaced with R^2 in Corollary 1 of the current paper, thereby reducing the computational complexity of the resulting message-passing algorithm. This reduction is significant, as ν can be dependent on the graph size in the worst case (see Figure 4).

As our **third contribution**, we investigate the important question of how the parameter R , which describes the maximum number of states for auxiliary variables, grows with the size of the graph, as measured by the total number of variables M in the corresponding MRF. As a result, we identify a sufficient condition on G that guarantees the overall polynomial run-time of Algorithm 1 in relation to the graph size in our target problem (Problem 1; see Theorem 2).

8. Related Works

Several previous works have addressed the problem of exact MAP inference with global factors in the context of SSVMs when optimizing for non-decomposable loss functions. Joachims [86] proposed an algorithm for a set of multivariate losses, including the F_β -loss. However, the presented idea applies only to a simple case, where the corresponding mapping F in (2) decomposes into *non-overlapping* components (e.g., unary potentials).

Similar ideas based on the introduction of auxiliary variables have been proposed [87,88] to modify the belief propagation algorithm according to a special form of high-order potentials. Specifically, for the case of binary-valued variables $y_i \in \{0, 1\}$, the authors focus on the *univariate cardinality potentials* $\eta(\mathbf{G}(\mathbf{y})) = \eta(\sum_i y_i)$. For the tasks of sequence tagging and constituency parsing, ref. [45,46] propose an exact inference algorithm for the slack scaling formulation that focuses on the *univariate* dissimilarity measures $G(\mathbf{y}) = \sum_t \mathbb{1}_1[y_t^* \neq y_t]$ and $\mathbf{G}(\mathbf{y}) = \#FP(\mathbf{y})$ (see Table 1 for details). In [69] the authors extrapolate this idea and provide a unified strategy to tackle multivariate and non-decomposable loss functions.

In the current paper, we build on the results in [69] and generalize the target problem (Problem 1) by increasing the range of admissible applications. More precisely, we replace the binary operation $\odot: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ corresponding to either a summation or multiplication in the previous objective $F(\mathbf{y}) \odot \eta(\mathbf{G}(\mathbf{y}))$ with a function $H: \mathbb{R} \times \mathbb{R}^P \rightarrow \mathbb{R}$, thereby allowing for more subtle interactions between the energy of the core model F and the sufficient statistics G according to $H(F(\mathbf{y}), G(\mathbf{y}))$. The increased flexibility, however, must be further restricted in order for a corresponding solution to be optimal. We have found that it is sufficient to impose a requirement on H to be non-decreasing in the first argument. Note that for the special case of the objective in [69], this requirement automatically holds.

Furthermore, the previous works could only guarantee polynomial run-time in cases where (a) the core model can be represented by a tree-shaped factor graph, and (b) the maximal degree of a variable node in the factor graph is bounded by a constant. The former excludes problems with cyclic dependencies and the latter rejects graphs with star-like

shapes. The corresponding idea for clique trees can handle cycles but suffers from a similar restriction on the maximal node degree being bounded. Here, we solve this problem by applying the graph transformation proposed in Section 3.2, effectively reducing the maximal node degree to $\nu = 3$. We note that a similar idea can be applied to factor graphs by replicating variable nodes and introducing constant factors. As a result, we improve the guarantee on the computational complexity by reducing the potentially unbounded parameter ν in the upper bound $O(M \cdot N^{\tau+1} \cdot R^{\nu-1})$ to $\nu \leq 3$.

In future work, we will consider applying our framework to the explanation task based on the technique of layer-wise relevance propagation [89] in graph neural networks, following in the spirit of [90,91].

9. Conclusions

Despite the high diversity in the range of existing applications, a considerable number of the underlying MAP problems share the unifying property that the information on the global variable interactions imposed by either a global factor or a global constraint can be locally propagated through the network by means of dynamic programming. By extending previous works, we presented a theoretical framework for efficient exact inference on models involving global factors with decomposable internal structures. At the heart of the presented framework is a constrained message-passing algorithm that always finds an optimal solution for our target problem in polynomial time. In particular, the performance of our approach does not explicitly depend on the graph form but rather on intrinsic properties such as the treewidth and the number of states of the auxiliary variables defined by the sufficient statistics of global interactions. The overall computational procedure is provably exact, and it has lower asymptotic bounds on the computational time complexity compared to previous works.

Author Contributions: Conceptualization, A.B.; Methodology, A.B.; Software, A.B.; Validation, A.B.; Formal analysis, A.B. and S.N.; Investigation, A.B.; Writing—original draft, A.B.; Writing—review & editing, A.B., S.N. and K.-R.M.; Funding acquisition, K.-R.M. All authors have read and agreed to the published version of the manuscript.

Funding: A.B. acknowledges support by the BASLEARN—TU Berlin/BASF Joint Lab for Machine Learning, cofinanced by TU Berlin and BASF SE. S.N. and K.-R.M. acknowledge support by the German Federal Ministry of Education and Research (BMBF) for BIFOLD under grants 01IS18025A and 01IS18037A. K.-R.M. was partly supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grants funded by the Korea government (MSIT) (no. 2019-0-00079, Artificial Intelligence Graduate School Program, Korea University and no. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation) and by the German Federal Ministry for Education and Research (BMBF) under grants 01IS14013B-E and 01GQ1115.

Data Availability Statement: The data used in the experimental part of this paper is available at <https://catalog ldc.upenn.edu>.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

LCS	Longest Common Subsequence
MAP	Maximum A Posteriori
MRF	Markov Random Field
SSVM	Structural Support Vector Machine

Appendix A. Proof of Theorem 1

Proof. We now show the correctness of the presented computations. For this purpose, we first provide a semantic interpretation of messages as follows. Let $C_i - C_j$ be an edge in a clique tree. We denote by $\mathcal{F}_{\prec(i-j)}$ the set of clique factors f_{C_k} of the mapping F on the C_i -th side of the tree, and by $\mathcal{G}_{\prec(i-j)}$ the corresponding set of clique factors of the mapping G . Furthermore, we denote by $\mathcal{V}_{\prec(i-j)}$ the set of all variables appearing on the C_i -th side but not in the sepset $C_i \cap C_j$. Intuitively, a message $\mu_{C_i \rightarrow C_j}^{l_i}(\mathbf{y}_{C_i \cap C_j})$ sent from clique C_i to C_j corresponds to the sum of all factors contained in $\mathcal{F}_{\prec(i-j)}$, which is maximized (for fixed values of $\mathbf{y}_{C_i \cap C_j}$ and l_i) over the variables in $\mathcal{V}_{\prec(i-j)}$ subject to the constraint $l_i = \sum_{g_{C_k} \in \mathcal{G}_{\prec(i-j)}} g_{C_k}(\mathbf{y}_{C_k})$. In other words, we define the following induction hypothesis:

$$\mu_{C_i \rightarrow C_j}^{l_i}(\mathbf{y}_{C_i \cap C_j}) = \max_{\mathcal{V}_{\prec(i-j)} : l_i = \sum_{g_{C_k} \in \mathcal{G}_{\prec(i-j)}} g_{C_k}(\mathbf{y}_{C_k})} \sum_{f_{C_k} \in \mathcal{F}_{\prec(i-j)}} f_{C_k}(\mathbf{y}_{C_k}). \tag{A1}$$

Now, consider an edge $(C_i - C_j)$ such that C_i is not a leaf. Let i_1, \dots, i_m be the neighboring cliques of C_i other than C_j . It follows from the running intersection property that $\mathcal{V}_{\prec(i-j)}$ is a disjoint union of $\mathcal{V}_{\prec(i_k-i)}$ for $k = 1, \dots, m$ and the variables $\mathbf{y}_{C_i \setminus C_j}$ eliminated at C_i itself. Similarly, $\mathcal{F}_{\prec(i-j)}$ is the disjoint union of the $\mathcal{F}_{\prec(i_k-i)}$ and $\{f_{C_i}\}$. Finally, $\mathcal{G}_{\prec(i-j)}$ is the disjoint union of the $\mathcal{G}_{\prec(i_k-i)}$ and $\{g_{C_i}\}$. In the following, we abbreviate the term $\mathcal{V}_{\prec(i_k-i)} : l_{i_k} = \sum_{g \in \mathcal{G}_{\prec(i_k-i)}} g$ describing a range of variables in $\mathcal{V}_{\prec(i_k-i)}$ subject to the corresponding equality constraint with respect to l_{i_k} by $\mathcal{V}_{\prec(i_k-i)} : l_{i_k}$. Thus, the right-hand side of Equation (A1) is equal to

$$\max_{\mathbf{y}_{C_i \setminus C_j}} \max_{\{l_{i_k}\}_{k=1}^m} \max_{\mathcal{V}_{\prec(i_1-i)} : l_{i_1}} \cdots \max_{\mathcal{V}_{\prec(i_m-i)} : l_{i_m}} \left(\sum_{f \in \mathcal{F}_{\prec(i_1-i)}} f \right) + \cdots + \left(\sum_{f \in \mathcal{F}_{\prec(i_m-i)}} f \right) + f_{C_i} \tag{A2}$$

where in the second max, we maximize over all configurations of $\{l_{i_k}\}_{k=1}^m$ subject to the constraint $\sum_{k=1}^m l_{i_k} = l_i - g_{C_i}(\mathbf{y}_{C_i})$. Since all the corresponding sets are disjoint, the term (A2) is equal to

$$\max_{\mathbf{y}_{C_i \setminus C_j}, \{l_{i_k}\}_{k=1}^m} f_{C_i} + \underbrace{\max_{\mathcal{V}_{\prec(i_1-i)} : l_{i_1}} \left(\sum_{f \in \mathcal{F}_{\prec(i_1-i)}} f \right)}_{\mu_{C_{i_1} \rightarrow C_i}^{l_{i_1}}(\mathbf{y}_{C_{i_1} \cap C_i})} + \cdots + \underbrace{\max_{\mathcal{V}_{\prec(i_m-i)} : l_{i_m}} \left(\sum_{f \in \mathcal{F}_{\prec(i_m-i)}} f \right)}_{\mu_{C_{i_m} \rightarrow C_i}^{l_{i_m}}(\mathbf{y}_{C_{i_m} \cap C_i})} \tag{A3}$$

where, again, the maximization over $\{l_{i_k}\}_{k=1}^m$ is subject to the constraint $\sum_{k=1}^m l_{i_k} = l_i - g_{C_i}(\mathbf{y}_{C_i})$. Using the induction hypothesis in the last expression, we obtain the right-hand side of Equation (7), thereby proving the claim in Equation (A1).

Now, look at Equation (9). By using Equation (A1) and considering that all the sets of variables and factors involved in different messages are disjoint, we can conclude that the computed values $\mu(l)$ correspond to the sum of all factors f for the mapping F over the variables in \mathbf{y} , which is maximized subject to the constraint $G(\mathbf{y}) = l$. Note that up until now, the proof is equivalent to that provided for Theorem 2 in the previous publication [69] because the message-passing step constrained via auxiliary variables is identical. Now, we use an additional requirement on H to ensure the optimality of the corresponding solution. Because H is non-decreasing in the first argument, by performing maximization over all values l according to Equation (10), we obtain the optimal value of Problem 1.

By inspecting the formula for the message passing in Equation (7), we can conclude that the corresponding operations can be performed in $O(M \cdot N^{\tau+1} \cdot R^{\nu-1})$ time, where ν denotes the maximal number of neighbors of any clique node C_i . First, the summation in Equation (7) involves $|ne(C_i)|$ terms, resulting in $|ne(C_i)| - 1$ summation operations.

Second, a maximization is performed first over $|C_i \setminus C_j|$ variables with a cost $N^{|C_i \setminus C_j|}$. This, however, is carried out for each configuration of $\mathbf{y}_{C_i \cap C_j}$, where $|C_i \setminus C_j| + |C_i \cap C_j| = |C_i| \leq \tau + 1$, resulting in $N^{\tau+1}$. Then, a maximization over $\{l_k\}$ costs an additional $R^{|\text{ne}(C_i)|-2}$. Together with the possible values for l , it yields $R^{\nu-1}$, where we upper bound $|\text{ne}(C_i)|$ by ν . Therefore, sending a message for all possible configurations of $(\mathbf{y}_{C_i \cap C_j}; l)$ on the edge $C_i - C_j$ costs $O(N^{\tau+1} \cdot (|\text{ne}(C_i)| - 1) \cdot R^{\nu-1})$ time. Finally, we need to carry out these operations for each edge $(i, j) \in E$ in the clique tree. The resulting cost can be estimated as follows: $\sum_{(i,j) \in E} N^{\tau+1} \cdot R^{\nu-1} \cdot (|\text{ne}(C_i)| - 1) = N^{\tau+1} \cdot R^{\nu-1} \sum_{(i,j) \in E} (|\text{ne}(C_i)| - 1) \leq N^{\tau+1} \cdot R^{\nu-1} \cdot |E| = N^{\tau+1} \cdot R^{\nu-1} \cdot (|V| - 1) \leq N^{\tau+1} \cdot R^{\nu-1} \cdot M$, where V denotes the set of clique nodes in the clique tree. Therefore, the total complexity is upper bounded by $O(M \cdot N^{\tau+1} \cdot R^{\nu-1})$. \square

Appendix B. Flowchart Diagram of Algorithm 1

We present a flowchart diagram for Algorithm 1 in Figure A1.

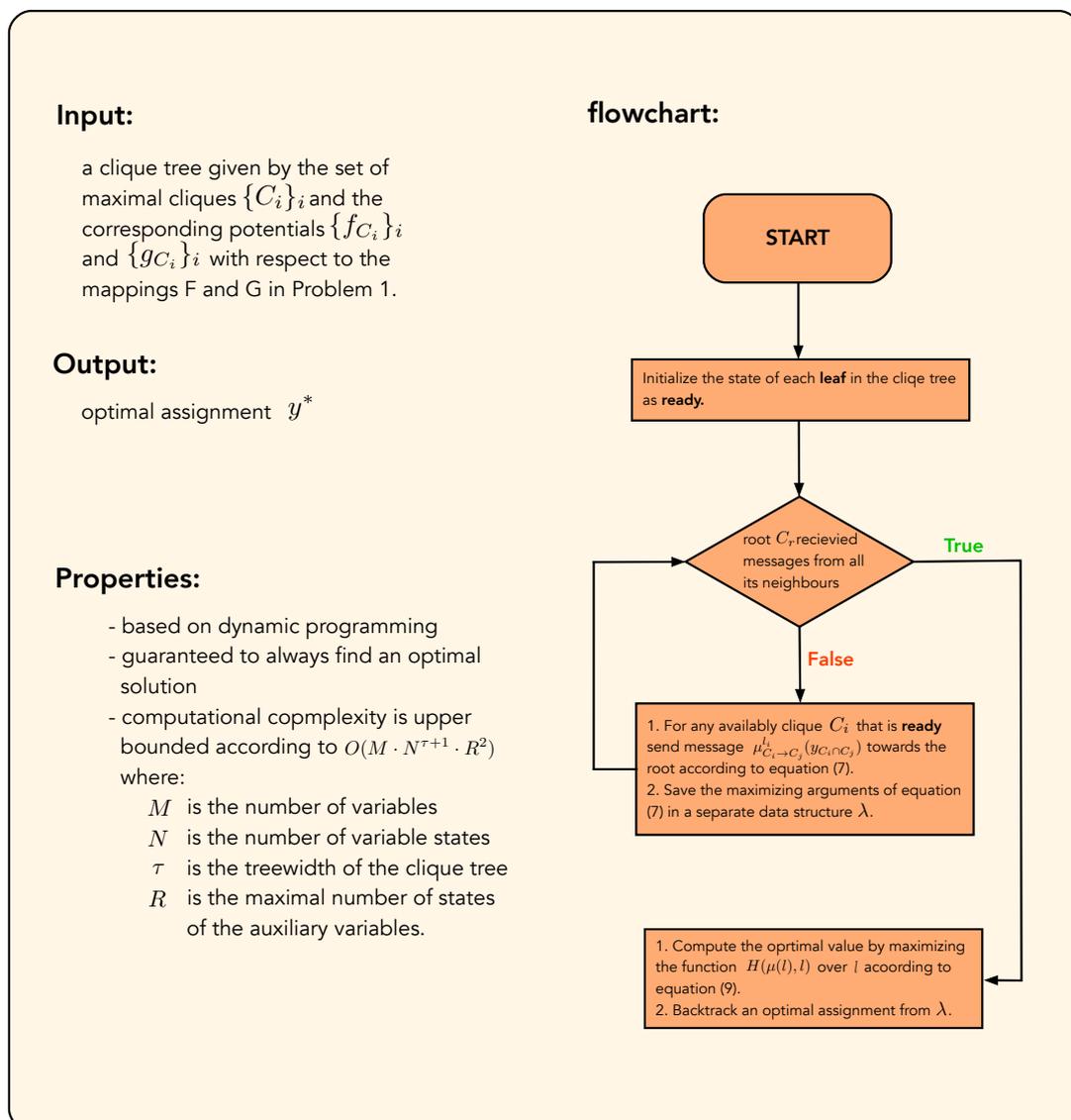


Figure A1. Flowchart diagram for Algorithm 1 for implementing a constrained message-passing scheme on clique trees. Algorithm 1 is guaranteed to always find an optimal solution in polynomial time. Its computational complexity is upper bounded according to $O(M \cdot N^{\tau+1} \cdot R^2)$.

Appendix C. Proof of Proposition 2

Proof. Assume we are given a clique tree with treewidth τ . This means that every node in the clique tree has at most $\tau + 1$ variables. Therefore, the number of all possible sepsets for a clique, which refers to the number of all variable combinations shared between two neighbors, is given by $2^{\tau+1} - 2$, where we exclude the empty set and the set containing all the variables in the corresponding clique.

Furthermore, we can deal with sepset duplicates by iteratively rearranging the edges in the clique tree such that for every sepset from the $2^{\tau+1} - 2$ possibilities, there is at most one duplicate providing an upper bound of $2^{\tau+2} - 4$ on the number of edges for each node. More precisely, we first choose a node with more than $2^{\tau+2} - 4$ neighbors as the root and then reshape the clique tree by propagating some of the duplicate edges (together with the corresponding subtrees) toward the leaves, as illustrated in Figure A2. The duplicate edges of nodes connected to the leaves of the clique tree can be reattached in a sequential manner similar to the example shown in Figure 4. Due to this procedure, the maximal number of duplicates for every sepset is upper bounded by 2. Multiplied by the maximal number of possible sepsets for each node, we obtain an upper bound on the number of neighbors in the reshaped clique tree given by $\nu \leq 2^{\tau+2} - 4$. \square

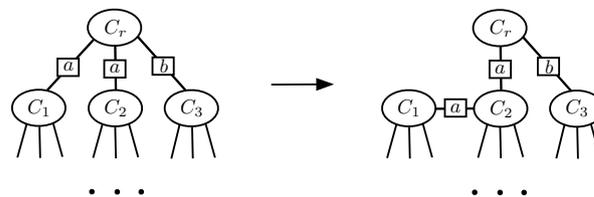


Figure A2. Illustration of the reshaping procedure for a clique tree in the case where the condition $\nu \leq 2^{\tau+2} - 4$ is violated. C_r is the root clique where a sepset a occurs at least two times. The number of neighbors of C_r can be reduced by removing the edge between C_1 and C_r and attaching C_1 to C_2 . In this way, we can ensure that every node has at most one duplicate for every possible sepset. Furthermore, this procedure preserves the running intersection property.

Appendix D. Proof of Theorem 2

Proof. We provide the proof by induction. Let C_1, \dots, C_K be the cliques of a corresponding instance of Problem 1. We now consider an arbitrary but fixed order of C_k for $k \in \{1, \dots, K\}$. We denote by R_i the number of states of a variable l_i , that is, R is given by $\max_i R_i$. As previously mentioned (see Equation (8)), an auxiliary variable l_i corresponds to the sum of potentials $g_{C_k}(y_{C_k})$ over all C_k in a subtree of which C_i is the root. This means that the number R_i is upper bounded by the image size of the corresponding sum function according to

$$R_i \leq \left| \sum_{k=1}^i \underbrace{g_{C_k}(\cdot)}_{\in [-i \cdot T, i \cdot T] \cap \mathbb{Z}} \right| \leq 2 \cdot i \cdot T \tag{A4}$$

where the corresponding values are in the set $[-i \cdot T, i \cdot T] \cap \mathbb{Z}$, defining our induction hypothesis. Using the induction hypothesis and the assumption $g_{C_k} \in [-T, T]$, it directly follows that the values for R_{i+1} are all in the set $[-(i+1) \cdot T, (i+1) \cdot T] \cap \mathbb{Z}$. The base case for R_1 holds due to the assumption of the theorem. Because $K \leq M$ always holds and the order of the considered cliques is arbitrary, we can conclude that R is upper bounded by $2 \cdot M \cdot T$, which is a polynomial in M . \square

References

1. Koller, D.; Friedman, N. *Probabilistic Graphical Models: Principles and Techniques-Adaptive Computation and Machine Learning*; The MIT Press: Cambridge, MA, USA, 2009.
2. Wainwright, M.J.; Jordan, M.I. Graphical Models, Exponential Families, and Variational Inference. *Found. Trends Mach. Learn.* **2008**, *1*, 1–305. [[CrossRef](#)]

3. Lafferty, J. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the ICML, Williamstown, MA, USA, 28 June–1 July 2001; pp. 282–289.
4. Kappes, J.H.; Andres, B.; Hamprecht, F.A.; Schnörr, C.; Nowozin, S.; Batra, D.; Kim, S.; Kausler, B.X.; Kröger, T.; Lellmann, J.; et al. A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems. *Int. J. Comput. Vis.* **2015**, *115*, 155–184. [[CrossRef](#)]
5. Bauer, A.; Nakajima, S.; Görnitz, N.; Müller, K.R. Partial Optimality of Dual Decomposition for MAP Inference in Pairwise MRFs. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, Naha, Okinawa, 16–18 April 2019; Volume 89, pp. 1696–1703.
6. Wainwright, M.J.; Jaakkola, T.S.; Willsky, A.S. MAP estimation via agreement on trees: Message-passing and linear programming. *IEEE Trans. Inf. Theory* **2005**, *51*, 3697–3717. [[CrossRef](#)]
7. Kolmogorov, V. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1568–1583. [[CrossRef](#)] [[PubMed](#)]
8. Kolmogorov, V.; Wainwright, M.J. On the Optimality of Tree-reweighted Max-product Message-passing. In Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, UK, 26–29 July 2005; pp. 316–323.
9. Sontag, D.; Meltzer, T.; Globerson, A.; Jaakkola, T.S.; Weiss, Y. Tightening LP Relaxations for MAP using Message Passing. In Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence, Helsinki, Finland, 9–12 July 2012.
10. Sontag, D. Approximate Inference in Graphical Models Using LP Relaxations. Ph.D. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA, 2010.
11. Sontag, D.; Globerson, A.; Jaakkola, T. Introduction to Dual Decomposition for Inference. In *Optimization for Machine Learning*; MIT Press: Cambridge, MA, USA, 2011.
12. Wang, J.; Yeung, S. A Compact Linear Programming Relaxation for Binary Sub-modular MRF. In Proceedings of the Energy Minimization Methods in Computer Vision and Pattern Recognition-10th International Conference, EMMCVPR 2015, Hong Kong, China, 13–16 January 2015; pp. 29–42.
13. Iii, H.D.; Marcu, D. Learning as search optimization: Approximate large margin methods for structured prediction. In Proceedings of the ICML, Bonn, Germany, 7–11 August 2005; pp. 169–176.
14. Sheng, L.; Binbin, Z.; Sixian, C.; Feng, L.; Ye, Z. Approximated Slack Scaling for Structural Support Vector Machines in Scene Depth Analysis. *Math. Probl. Eng.* **2013**, *2013*, 817496.
15. Kulesza, A.; Pereira, F. Structured Learning with Approximate Inference. In Proceedings of the 20th NIPS, Vancouver British Columbia, Canada, 3–6 December 2007; pp. 785–792.
16. Rush, A.M.; Collins, M.J. A Tutorial on Dual Decomposition and Lagrangian Relaxation for Inference in Natural Language Processing. *J. Artif. Intell. Res.* **2012**, *45*, 305–362. [[CrossRef](#)]
17. Bodenstein, N.; Dunlop, A.; Hall, K.B.; Roark, B. Beam-Width Prediction for Efficient Context-Free Parsing. In Proceedings of the 49th ACL, Portland, OR, USA, 19–24 June 2011; pp. 440–449.
18. Ratliff, N.D.; Bagnell, J.A.; Zinkevich, M. (Approximate) Subgradient Methods for Structured Prediction. In Proceedings of the 11th AISTATS, San Juan, Puerto Rico, 21–24 March 2007; pp. 380–387.
19. Lim, Y.; Jung, K.; Kohli, P. Efficient Energy Minimization for Enforcing Label Statistics. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1893–1899. [[CrossRef](#)] [[PubMed](#)]
20. Ranjbar, M.; Vahdat, A.; Mori, G. Complex loss optimization via dual decomposition. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2304–2311.
21. Komodakis, N.; Paragios, N. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009; pp. 2985–2992.
22. Boykov, Y.; Veksler, O. Graph Cuts in Vision and Graphics: Theories and Applications. In *Handbook of Mathematical Models in Computer Vision*; 2006; pp. 79–96. Available online: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=279e0f5d885110c173bb86d37c997becf198651b> (accessed on 16 May 2017).
23. Kolmogorov, V.; Zabih, R. What Energy Functions Can Be Minimized via Graph Cuts? *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 147–159. [[CrossRef](#)]
24. Hurley, B.; O’Sullivan, B.; Allouche, D.; Katsirelos, G.; Schiex, T.; Zytnicki, M.; de Givry, S. Multi-language evaluation of exact solvers in graphical model discrete optimization. *Constraints* **2016**, *21*, 413–434. [[CrossRef](#)]
25. Haller, S.; Swoboda, P.; Savchynskyy, B. Exact MAP-Inference by Confining Combinatorial Search With LP Relaxation. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 6581–6588.
26. Savchynskyy, B.; Kappes, J.H.; Swoboda, P.; Schnörr, C. Global MAP-Optimality by Shrinking the Combinatorial Search Area with Convex Relaxation. In Proceedings of the Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 1950–1958.
27. Kappes, J.H.; Speth, M.; Reinelt, G.; Schnörr, C. Towards Efficient and Exact MAP-Inference for Large Scale Discrete Computer Vision Problems via Combinatorial Optimization. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1752–1758.

28. Forney, G.D. The Viterbi algorithm. *Proc. IEEE* **1973**, *61*, 268–278. [[CrossRef](#)]
29. Tarlow, D.; Givoni, I.E.; Zemel, R.S. HOP-MAP: Efficient message passing with high order potentials. In Proceedings of the 13th AISTATS, Sardinia, Italy, 13–15 May 2010.
30. McAuley, J.J.; Caetano, T.S. Faster Algorithms for Max-Product Message-Passing. *J. Mach. Learn. Res.* **2011**, *12*, 1349–1388.
31. Younger, D.H. Recognition and Parsing of Context-Free Languages in Time n^3 . *Inf. Control* **1967**, *10*, 189–208. [[CrossRef](#)]
32. Klein, D.; Manning, C.D. A* Parsing: Fast Exact Viterbi Parse Selection. In Proceedings of the HLT-NAACL, Edmonton, AB, Canada, 31 May 2003; pp. 119–126.
33. Gupta, R.; Diwan, A.A.; Sarawagi, S. Efficient inference with cardinality-based clique potentials. In Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, OR, USA, 20–24 June 2007; pp. 329–336.
34. Kolmogorov, V.; Boykov, Y.; Roth, C. Applications of parametric maxflow in computer vision. In Proceedings of the IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, 14–20 October 2007; pp. 1–8.
35. McAuley, J.J.; Caetano, T.S. Exploiting Within-Clique Factorizations in Junction-Tree Algorithms. In Proceedings of the 13th AISTATS, Sardinia, Italy, 13–15 May 2010; pp. 525–532.
36. Bodlaender, H. A Tourist Guide Through Treewidth. *Acta Cybern.* **1993**, *11*, 1–22.
37. Chandrasekaran, V.; Srebro, N.; Harsha, P. Complexity of Inference in Graphical Models. In Proceedings of the 24th Conference in Artificial Intelligence, Helsinki, Finland, 9–12 July 2008; pp. 70–78.
38. Taskar, B.; Guestrin, C.; Koller, D. Max-Margin Markov Networks. In Proceedings of the 16th NIPS, Whistler, BC, Canada, 9–11 December 2003; pp. 25–32.
39. Tsochantaris, I.; Joachims, T.; Hofmann, T.; Altun, Y. Large Margin Methods for Structured and Interdependent Output Variables. *J. Mach. Learn. Res.* **2005**, *6*, 1453–1484.
40. Joachims, T.; Hofmann, T.; Yue, Y.; Yu, C.N. Predicting Structured Objects with Support Vector Machines. *Commun. ACM Res. Highlight* **2009**, *52*, 97–104. [[CrossRef](#)]
41. Sarawagi, S.; Gupta, R. Accurate max-margin training for structured output spaces. In Proceedings of the 25th ICML, Helsinki, Finland, 5–9 July 2008; pp. 888–895.
42. Taskar, B.; Klein, D.; Collins, M.; Koller, D.; Manning, C.D. Max-Margin Parsing. In Proceedings of the EMNLP, Barcelona, Spain, 25–26 July 2004; pp. 1–8.
43. Nam John Yu, C.; Joachims, T. Learning Structural SVMs with Latent Variables. In Proceedings of the ICML, Montreal, QC, Canada, 14–18 June 2009; pp. 1169–1176.
44. Bakir, G.; Hoffman, T.; Schölkopf, B.; Smola, A.J.; Taskar, B.; Vishwanathan, S.V.N. *Predicting Structured Data*; The MIT Press: Cambridge, MA, USA, 2007.
45. Bauer, A.; Görnitz, N.; Biegler, F.; Müller, K.R.; Kloft, M. Efficient Algorithms for Exact Inference in Sequence Labeling SVMs. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 870–881. [[CrossRef](#)]
46. Bauer, A.; Braun, M.L.; Müller, K.R. Accurate Maximum-Margin Training for Parsing With Context-Free Grammars. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 44–56. [[CrossRef](#)] [[PubMed](#)]
47. Bauer, A.; Nakajima, S.; Görnitz, N.; Müller, K.R. Optimizing for Measure of Performance in Max-Margin Parsing. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 2680–2684. [[CrossRef](#)]
48. McAllester, D. Generalization bounds and consistency for structured labeling. In *Proceedings of the Predicting Structured Data*; Gökhan, B.H., Hofmann, T., Schölkopf, B., Smola, A.J., Taskar, B., Vishwanathan, S.V.N., Eds.; The MIT Press: Cambridge, MA, USA, 2006; pp. 247–262.
49. McAllester, D.A.; Keshet, J. Generalization Bounds and Consistency for Latent Structural Probit and Ramp Loss. In Proceedings of the 25th NIPS, Granada, Spain, 12–15 December 2011; pp. 2205–2212.
50. London, B.; Huang, B.; Getoor, L. Stability and Generalization in Structured Prediction. *J. Mach. Learn. Res.* **2016**, *17*, 1–52.
51. Ranjbar, M.; Mori, G.; Wang, Y. Optimizing Complex Loss Functions in Structured Prediction. In Proceedings of the 11th ECCV, Heraklion, Crete, Greece, 5–11 September 2010; pp. 580–593.
52. Rätsch, G.; Sonnenburg, S. Large Scale Hidden Semi-Markov SVMs. In Proceedings of the 19 NIPS, Barcelona, Spain, 9 December 2007; pp. 1161–1168.
53. Tarlow, D.; Zemel, R.S. Structured Output Learning with High Order Loss Functions. In Proceedings of the 15th AISTATS, La Palma, Canary Islands, Spain, 21–23 April 2012; pp. 1212–1220.
54. Taskar, B.; Chatalbashev, V.; Koller, D.; Guestrin, C. Learning structured prediction models: A large margin approach. In Proceedings of the 22nd ICML, Bonn, Germany, 7–11 August 2005; pp. 896–903.
55. Finley, T.; Joachims, T. Training Structural SVMs when Exact Inference is Intractable. In Proceedings of the 25th ICML, Helsinki, Finland, 5–9 July 2008; pp. 304–311.
56. Meshi, O.; Sontag, D.; Jaakkola, T.S.; Globerson, A. Learning Efficiently with Approximate Inference via Dual Losses. In Proceedings of the 27th ICML, Haifa, Israel, 21–24 June 2010; pp. 783–790.
57. Balamurugan, P.; Shevade, S.K.; Sundararajan, S. A Simple Label Switching Algorithm for Semisupervised Structural SVMs. *Neural Comput.* **2015**, *27*, 2183–2206. [[CrossRef](#)]
58. Shevade, S.K.; Balamurugan, P.; Sundararajan, S.; Keerthi, S.S. A Sequential Dual Method for Structural SVMs. In Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, Mesa, AZ, USA, 28–30 April 2011; pp. 223–234.

59. Taskar, B.; Lacoste-Julien, S.; Jordan, M.I. Structured Prediction, Dual Extragradient and Bregman Projections. *J. Mach. Learn. Res.* **2006**, *7*, 1627–1653.
60. Nowozin, S.; Gehler, P.V.; Jancsary, J.; Lampert, C.H. *Advanced Structured Prediction*; The MIT Press: Cambridge, MA, USA, 2014.
61. Martins, A.F.T.; Figueiredo, M.A.T.; Aguiar, P.M.Q.; Smith, N.A.; Xing, E.P. An Augmented Lagrangian Approach to Constrained MAP Inference. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, WA, USA, 28 June–2 July 2011*; Getoor, L., Scheffer, T., Eds.; Omnipress: Paraskevi, Greece, 2011; pp. 169–176.
62. Batra, D.; Yadollahpour, P.; Guzmán-Rivera, A.; Shakhnarovich, G. Diverse M-Best Solutions in Markov Random Fields. In *Proceedings of the Computer Vision-ECCV 2012-12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Part V*; pp. 1–16.
63. Guzmán-Rivera, A.; Kohli, P.; Batra, D. DivMCuts: Faster Training of Structural SVMs with Diverse M-Best Cutting-Planes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, 29 April–1 May 2013*; pp. 316–324.
64. Joachims, T.; Finley, T.; Yu, C.N. Cutting-Plane Training of Structural SVMs. *Mach. Learn.* **2009**, *77*, 27–59. [[CrossRef](#)]
65. Kelley, J.E. The Cutting-Plane Method for Solving Convex Programs. *J. Soc. Ind. Appl. Math.* **1960**, *8*, 703–712. [[CrossRef](#)]
66. Lacoste-Julien, S.; Jaggi, M.; Schmidt, M.W.; Pletscher, P. Block-Coordinate Frank-Wolfe Optimization for Structural SVMs. In *Proceedings of the 30th ICML, Atlanta, GA, USA, 16–21 June 2013*; pp. 53–61.
67. Teo, C.H.; Vishwanathan, S.V.N.; Smola, A.J.; Le, Q.V. Bundle Methods for Regularized Risk Minimization. *J. Mach. Learn. Res.* **2010**, *11*, 311–365.
68. Smola, A.J.; Vishwanathan, S.V.N.; Le, Q.V. Bundle Methods for Machine Learning. In *Proceedings of the 21st NIPS, Vancouver, BC, Canada, 7–8 December 2007*; pp. 1377–1384.
69. Bauer, A.; Nakajima, S.; Müller, K.R. Efficient Exact Inference with Loss Augmented Objective in Structured Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2566–2579. [[CrossRef](#)]
70. Bishop, C.M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*; Springer, Inc.: Secaucus, NJ, USA, 2006.
71. Lauritzen, S.L.; Spiegelhalter, D.J. Chapter Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. In *Readings in Uncertain Reasoning*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1990; pp. 415–448.
72. Johnson, M. PCFG Models of Linguistic Tree Representations. *Comput. Linguist.* **1998**, *24*, 613–632.
73. Heemskerk, J.S. A Probabilistic Context-free Grammar for Disambiguation in Morphological Parsing. In *Proceedings of the EACL, Utrecht, The Netherlands, 19–23 April 1993*; pp. 183–192.
74. Charniak, E. Statistical Parsing with a Context-Free Grammar and Word Statistics. In *Proceedings of the 40th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence, Providence, RI, USA, 27–31 July 1997*; pp. 598–603.
75. Rabiner, L.R. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286. [[CrossRef](#)]
76. Koller, D. Probabilistic Relational Models. In *Proceedings of the Inductive Logic Programming, 9th International Workshop, ILP-99, Bled, Slovenia, 24–27 June 1999*; pp. 3–13.
77. Taskar, B.; Abbeel, P.; Koller, D. Discriminative Probabilistic Models for Relational Data. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence, Edmonton, AB, Canada, 11–15 July 2013*.
78. Richardson, M.; Domingos, P.M. Markov logic networks. *Mach. Learn.* **2006**, *62*, 107–136. [[CrossRef](#)]
79. Komodakis, N.; Paragios, N.; Tziritas, G. MRF Energy Minimization and Beyond via Dual Decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 531–552. [[CrossRef](#)]
80. Everingham, M.; Eslami, S.M.A.; Gool, L.J.V.; Williams, C.K.I.; Winn, J.M.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [[CrossRef](#)]
81. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. Bleu: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002*; pp. 311–318.
82. He, T.; Chen, J.; Ma, L.; Gui, Z.; Li, F.; Shao, W.; Wang, Q. ROUGE-C: A Fully Automated Evaluation Method for Multi-document Summarization. In *Proceedings of the 2008 IEEE International Conference on Granular Computing, GrC 2008, Hangzhou, China, 26–28 August 2008*; pp. 269–274.
83. Manning, C.D. Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? In *Computational Linguistics and Intelligent Text Processing, Proceedings of the 12th International Conference, CICLing 2011, Tokyo, Japan, 20–26 February 2011; Proceedings, Part I; Lecture Notes in Computer Science*; Gelbukh, A.F., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6608, pp. 171–189.
84. Tongcham, S.; Sornlertlamvanich, V.; Isahara, H. Experiments in Base-NP Chunking and Its Role in Dependency Parsing for Thai. In *Proceedings of the 22nd COLING, Manchester, UK, 18–22 August 2008*; pp. 123–126.
85. Marcus, M.P.; Kim, G.; Marcinkiewicz, M.A.; MacIntyre, R.; Bies, A.; Ferguson, M.; Katz, K.; Schasberger, B. The Penn Treebank: Annotating Predicate Argument Structure. In *Proceedings of the Human Language Technology, Plainsboro, NJ, USA, 8–11 March 1994*.
86. Joachims, T. A Support Vector Method for Multivariate Performance Measures. In *Proceedings of the 22nd ICML, Bonn, Germany, 7–11 August 2005*; pp. 377–384.

87. Tarlow, D.; Swersky, K.; Zemel, R.S.; Adams, R.P.; Frey, B.J. Fast Exact Inference for Recursive Cardinality Models. In Proceedings of the 28th UAI, Catalina Island, CA, USA, 14–18 August 2012.
88. Mezuman, E.; Tarlow, D.; Globerson, A.; Weiss, Y. Tighter Linear Program Relaxations for High Order Graphical Models. In Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI), Bellevue, WA, USA, 11–15 July 2013.
89. Kohlbrenner, M.; Bauer, A.; Nakajima, S.; Binder, A.; Samek, W.; Lapuschkin, S. Towards Best Practice in Explaining Neural Network Decisions with LRP. In Proceedings of the 2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, UK, 19–24 July 2020; pp. 1–7.
90. Schnake, T.; Eberle, O.; Lederer, J.; Nakajima, S.; Schütt, K.T.; Müller, K.; Montavon, G. Higher-Order Explanations of Graph Neural Networks via Relevant Walks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7581–7596. [[CrossRef](#)] [[PubMed](#)]
91. Xiong, P.; Schnake, T.; Montavon, G.; Müller, K.R.; Nakajima, S. Efficient Computation of Higher-Order Subgraph Attribution via Message Passing. In Proceedings of the 39th International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; Volume 162, pp. 24478–24495.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.