Structure Regularization for Structured Prediction

Xu Sun[*†]
∗MOE Key Laboratory of Computational Linguistics, Peking University
†School of Electronics Engineering and Computer Science, Peking University
xusun@pku.edu.cn

Abstract

While there are many studies on weight regularization, the study on structure regularization is rare. Many existing systems on structured prediction focus on increasing the level of structural dependencies within the model. However, this trend could have been misdirected, because our study suggests that complex structures are actually harmful to generalization ability in structured prediction. To control structure-based overfitting, we propose a structure regularization framework via *structure decomposition*, which decomposes training samples into mini-samples with simpler structures, deriving a model with better generalization power. We show both theoretically and empirically that structure regularization can effectively control overfitting risk and lead to better accuracy. As a by-product, the proposed method can also substantially accelerate the training speed. The method and the theoretical results can apply to general graphical models with arbitrary structures. Experiments on well-known tasks demonstrate that our method can easily beat the benchmark systems on those highly-competitive tasks, achieving state-of-the-art accuracies yet with substantially faster training speed.

1 Introduction

Structured prediction models are popularly used to solve structure dependent problems in a wide variety of application domains including natural language processing, bioinformatics, speech recognition, and computer vision. Recently, many existing systems on structured prediction focus on increasing the level of structural dependencies within the model. We argue that this trend could have been misdirected, because our study suggests that complex structures are actually harmful to model accuracy. While it is obvious that intensive structural dependencies can effectively incorporate structural information, it is less obvious that intensive structural dependencies have a drawback of increasing the generalization risk, because more complex structures are easier to suffer from overfitting. Since this type of overfitting is caused by structure complexity, it can hardly be solved by ordinary regularization methods such as L_2 and L_1 regularization schemes, which is only for controlling weight complexity.

To deal with this problem, we propose a simple structure regularization solution based on *tag structure decomposition*. The proposed method decomposes each training sample into multiple mini-samples with simpler structures, deriving a model with better generalization power. The proposed method is easy to implement, and it has several interesting properties: (1) We show both theoretically and empirically that the proposed method can reduce the overfit risk. (2) Keeping the convexity of the objective function: a convex function with a structure regularizer is still convex. (3) No conflict with the weight regularization: we can apply structure regularization together with weight regularization. (4) Accelerating the convergence rate in training. (5) This method can be used for different types of models, including CRFs [6] and perceptrons [3].

The term *structural regularization* has been used in prior work for regularizing *structures of features*, including spectral regularization [1], regularizing feature structures for classifiers [23], and many

recent studies on structured sparsity in structured prediction scenarios [13, 9], via adopting mixed norm regularization [11], *Group Lasso* [25], and posterior regularization [5]. Compared with those prior work, we emphasize that our proposal on tag structure regularization is novel. This is because the term *structure* in all of the aforementioned work refers to *structures of feature space*, which is substantially different compared with our proposal on regularizing tag structures (interactions among tags).

There are other related studies, including the studies of [20] and [12] on piecewise/decomposed training methods, and the study of [22] on a "lookahead" learning method. Our work differs from [20, 12, 22] mainly because our work is built on a regularization framework, with arguments and justifications on reducing generalization risk and for better accuracy. Also, our method and the theoretical results can fit general graphical models with arbitrary structures, and the detailed algorithm is quite different. On generalization risk analysis, related studies include [2, 14] on non-structured classification and [21, 8, 7] on structured classification.

To the best of our knowledge, this is the first theoretical result on quantifying the relation between structure complexity and the generalization risk in structured prediction, and this is also the first proposal on structure regularization via regularizing tag-interactions. The contributions of this work[1] are two-fold:

- On the methodology side, we propose a structure regularization framework for structured prediction. We show both theoretically and empirically that the proposed method can effectively reduce the overfitting risk, and at the same time accelerate the convergence rate in training. Our method and the theoretical analysis do *not* make assumptions based on specific structures. In other words, the method and the theoretical results can apply to graphical models with arbitrary structures, including linear chains, trees, and general graphs.
- On the application side, for several important natural language processing tasks, our simple method can easily beat the benchmark systems on those highly-competitive tasks, achieving record-breaking accuracies as well as substantially faster training speed.

2 Structure Regularization

A graph of observations (even with arbitrary structures) can be indexed and be denoted by using an indexed sequence of observations $\boldsymbol{O} = \{o_1, \ldots, o_n\}$. We use the term *sample* to denote $\boldsymbol{O} = \{o_1, \ldots, o_n\}$. For example, in natural language processing, a sample may correspond to a sentence of n words with dependencies of tree structures (e.g., in syntactic parsing). For simplicity in analysis, we assume all samples have n observations (thus n tags). In a typical setting of structured prediction, all the n tags have inter-dependencies via connecting each Markov dependency between neighboring tags. Thus, we call n as *tag structure complexity* or simply *structure complexity* below.

A sample is converted to an indexed sequence of feature vectors $\boldsymbol{x} = \{\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)}\}$, where $\boldsymbol{x}_{(k)} \in \mathcal{X}$ is of the dimension d and corresponds to the local features extracted from the position/index k. We can use an $n \times d$ matrix to represent $\boldsymbol{x} \in \mathcal{X}^n$. Let $\mathcal{Z} = (\mathcal{X}^n, \mathcal{Y}^n)$ and let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{Z}$ denote a sample in the training data. Suppose a training set is $S = \{\boldsymbol{z}_1 = (\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, \boldsymbol{z}_m = (\boldsymbol{x}_m, \boldsymbol{y}_m)\}$, with size m, and the samples are drawn i.i.d. from a distribution D which is unknown. A learning algorithm is a function $G : \mathcal{Z}^m \mapsto \mathcal{F}$ with the function space $\mathcal{F} \subset \{\mathcal{X}^n \mapsto \mathcal{Y}^n\}$, i.e., G maps a training set S to a function $G_S : \mathcal{X}^n \mapsto \mathcal{Y}^n$. We suppose G is symmetric with respect to S, so that G is independent on the order of S.

Structural dependencies among tags are the major difference between structured prediction and non-structured classification. For the latter case, a local classification of g based on a position k can be expressed as $g(\boldsymbol{x}_{(k-a)}, \ldots, \boldsymbol{x}_{(k+a)})$, where the term $\{\boldsymbol{x}_{(k-a)}, \ldots, \boldsymbol{x}_{(k+a)}\}$ represents a local window. However, for structured prediction, a local classification on a position depends on the whole input $\boldsymbol{x} = \{\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)}\}$ rather than a local window, due to the nature of structural dependencies among tags (e.g., graphical models like CRFs). Thus, in structured prediction a local classification on k should be denoted as $g(\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)}, k)$. To simplify the notation, we define

$$g(\boldsymbol{x}, k) \triangleq g(\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)}, k)$$

[1] See the code at `http://klcl.pku.edu.cn/member/sunxu/code.htm`

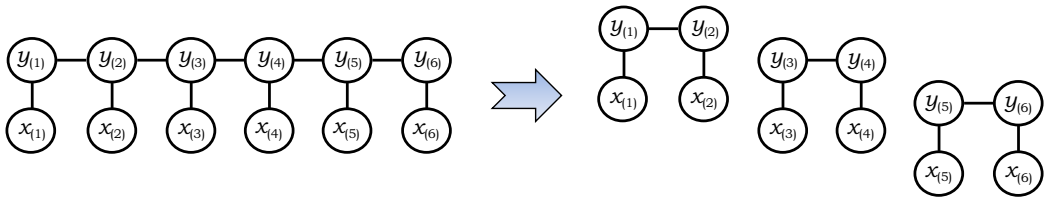

Figure 1: An illustration of structure regularization in simple linear chain case, which decompose a training sample $\boldsymbol{z}$ with structure complexity 6 into three mini-samples with structure complexity 2. Structure regularization can apply to more general graphs with arbitrary dependencies.

We define *point-wise cost function* $c : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}^+$ as $c[G_S(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}]$, which measures the cost on a position k by comparing $G_S(\boldsymbol{x}, k)$ and the gold-standard tag $\boldsymbol{y}_{(k)}$, and we introduce the point-wise loss as

$$\ell(G_S, \boldsymbol{z}, k) \triangleq c[G_S(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}]$$

Then, we define *sample-wise cost function* $C : \mathcal{Y}^n \times \mathcal{Y}^n \mapsto \mathbb{R}^+$, which is the cost function with respect to a whole sample, and we introduce the sample-wise loss as

$$\mathcal{L}(G_S, \boldsymbol{z}) \triangleq C[G_S(\boldsymbol{x}), \boldsymbol{y}] = \sum_{k=1}^{n} \ell(G_S, \boldsymbol{z}, k) = \sum_{k=1}^{n} c[G_S(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}]$$

Given G and a training set S, what we are most interested in is the *generalization risk* in structured prediction (i.e., expected average loss) [21, 8]:

$$R(G_S) = \mathbb{E}_{\boldsymbol{z}}\Big[\frac{\mathcal{L}(G_S, \boldsymbol{z})}{n}\Big]$$

Since the distribution D is unknown, we have to estimate $R(G_S)$ by using the *empirical risk*:

$$R_e(G_S) = \frac{1}{mn} \sum_{i=1}^{m} \mathcal{L}(G_S, \boldsymbol{z}_i) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{k=1}^{n} \ell(G_S, \boldsymbol{z}_i, k)$$

To state our theoretical results, we must describe several quantities and assumptions following prior work [2, 14]. We assume a simple real-valued structured prediction scheme such that the class predicted on position k of $\boldsymbol{x}$ is the sign of $G_S(\boldsymbol{x}, k) \in \mathcal{D}$.[2] Also, we assume the point-wise cost function c_τ is convex and τ-*smooth* such that $\forall y_1, y_2 \in \mathcal{D}, \forall y^* \in \mathcal{Y}$

$$|c_\tau(y_1, y^*) - c_\tau(y_2, y^*)| \leq \tau |y_1 - y_2| \tag{1}$$

Also, we use a value ρ to quantify the bound of $|G_S(\boldsymbol{x}, k) - G_{S^{\setminus i}}(\boldsymbol{x}, k)|$ while changing a single sample (with size $n' \leq n$) in the training set with respect to the structured input $\boldsymbol{x}$. This ρ-*admissible* assumption can be formulated as $\forall k$,

$$|G_S(\boldsymbol{x}, k) - G_{S^{\setminus i}}(\boldsymbol{x}, k)| \leq \rho ||G_S - G_{S^{\setminus i}}||_2 \cdot ||\boldsymbol{x}||_2 \tag{2}$$

where $\rho \in \mathbb{R}^+$ is a value related to the design of algorithm G.

2.1 Structure Regularization

Most existing regularization techniques are for regularizing model weights/parameters (e.g., a representative regularizer is the Gaussian regularizer or so called L_2 regularizer), and we call such regularization techniques as *weight regularization*.

Definition 1 (Weight regularization) *Let $N_\lambda : \mathcal{F} \mapsto \mathbb{R}^+$ be a weight regularization function on $\mathcal{F}$ with regularization strength λ, the structured classification based objective function with general weight regularization is as follows:*

$$R_\lambda(G_S) \triangleq R_e(G_S) + N_\lambda(G_S) \tag{3}$$

[2] Many popular structured prediction models have a convex and real-valued cost function (e.g., CRFs).

Algorithm 1 Training with structure regularization

Input: model weights $\boldsymbol{w}$, training set S, structure regularization strength α
repeat
  $S' \leftarrow \emptyset$
  for $i = 1 \rightarrow m$ **do**
    Randomly decompose $\boldsymbol{z}_i \in S$ into mini-samples $N_\alpha(\boldsymbol{z}_i) = \{\boldsymbol{z}_{(i,1)}, \ldots, \boldsymbol{z}_{(i,\alpha)}\}$
    $S' \leftarrow S' \cup N_\alpha(\boldsymbol{z}_i)$
  end for
  for $i = 1 \rightarrow |S'|$ **do**
    Sample $\boldsymbol{z}'$ uniformly at random from S', with gradient $\nabla g_{\boldsymbol{z}'}(\boldsymbol{w})$
    $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \nabla g_{\boldsymbol{z}'}(\boldsymbol{w})$
  end for
until Convergence
return $\boldsymbol{w}$

While weight regularization is normalizing model weights, the proposed structure regularization method is normalizing the structural complexity of the training samples. As illustrated in Figure 1, our proposal is based on *tag structure decomposition*, which can be formally defined as follows:

Definition 2 (Structure regularization) *Let $N_\alpha : \mathcal{F} \mapsto \mathcal{F}$ be a structure regularization function on $\mathcal{F}$ with regularization strength α with $1 \leq \alpha \leq n$, the structured classification based objective function with structure regularization is as follows*[3]*:*

$$R_\alpha(G_S) \triangleq R_e[G_{N_\alpha(S)}] = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{\alpha} \mathcal{L}[G_{S'}, \boldsymbol{z}_{(i,j)}] = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{\alpha} \sum_{k=1}^{n/\alpha} \ell[G_{S'}, \boldsymbol{z}_{(i,j)}, k] \tag{4}$$

where $N_\alpha(\boldsymbol{z}_i)$ randomly splits $\boldsymbol{z}_i$ into α mini-samples $\{\boldsymbol{z}_{(i,1)}, \ldots, \boldsymbol{z}_{(i,\alpha)}\}$, so that the mini-samples have a distribution on their sizes (structure complexities) with the expected value $n' = n/\alpha$. Thus, we get

$$S' = \{\underbrace{\boldsymbol{z}_{(1,1)}, z_{(1,2)}, \ldots, \boldsymbol{z}_{(1,\alpha)}}_{\alpha}, \ldots, \underbrace{\boldsymbol{z}_{(m,1)}, \boldsymbol{z}_{(m,2)}, \ldots, \boldsymbol{z}_{(m,\alpha)}}_{\alpha}\} \tag{5}$$

with $m\alpha$ mini-samples with expected structure complexity n/α. We can denote S' more compactly as $S' = \{\boldsymbol{z}'_1, \boldsymbol{z}'_2, \ldots, \boldsymbol{z}'_{m\alpha}\}$ and $R_\alpha(G_S)$ can be simplified as

$$R_\alpha(G_S) \triangleq \frac{1}{mn} \sum_{i=1}^{m\alpha} \mathcal{L}(G_{S'}, \boldsymbol{z}'_i) = \frac{1}{mn} \sum_{i=1}^{m\alpha} \sum_{k=1}^{n/\alpha} \ell[G_{S'}, \boldsymbol{z}'_i, k] \tag{6}$$

When the structure regularization strength $\alpha = 1$, we have $S' = S$ and $R_\alpha = R_e$. The structure regularization algorithm (with the stochastic gradient descent setting) is summarized in Algorithm 1. Recall that $\boldsymbol{x} = \{\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)}\}$ represents feature vectors. Thus, it should be emphasized that the decomposition of $\boldsymbol{x}$ is the decomposition of the feature vectors, not the original observations. Actually the decomposition of the feature vectors is more convenient and has no information loss — decomposing observations needs to regenerate features and may lose some features.

The structure regularization has no conflict with the weight regularization, and the structure regularization can be applied together with the weight regularization.

Definition 3 (Structure & weight regularization) *By combining structure regularization in Definition 2 and weight regularization in Definition 1, the structured classification based objective function is as follows:*

$$R_{\alpha,\lambda}(G_S) \triangleq R_\alpha(G_S) + N_\lambda(G_S) \tag{7}$$

When $\alpha = 1$, we have $R_{\alpha,\lambda} = R_e(G_S) + N_\lambda(G_S) = R_\lambda$.

Like existing weight regularization methods, currently our structure regularization is only for the training stage. Currently we do not use structure regularization in the test stage.

[3]The notation N is overloaded here. For clarity throughout, N with subscript λ refers to weight regularization function, and N with subscript α refers to structure regularization function.

2.2 Reduction of Generalization Risk

In contrast to the simplicity of the algorithm, the theoretical analysis is quite technical. In this paper we only describe the major theoretical result. Detailed analysis and proofs are given in the full version of this work [16].

Theorem 4 (Generalization vs. structure regularization) *Let the structured prediction objective function of G be penalized by structure regularization with factor $\alpha \in [1, n]$ and L_2 weight regularization with factor λ, and the penalized function has a minimizer f:*

$$f = \operatorname*{argmin}_{g \in \mathcal{F}} R_{\alpha,\lambda}(g) = \operatorname*{argmin}_{g \in \mathcal{F}} \Big(\frac{1}{mn} \sum_{j=1}^{m\alpha} \mathcal{L}_\tau(g, \boldsymbol{z}'_j) + \frac{\lambda}{2} ||g||_2^2 \Big) \tag{8}$$

Assume the point-wise loss ℓ_τ is convex and differentiable, and is bounded by $\ell_\tau(f, \boldsymbol{z}, k) \leq \gamma$. Assume $f(\boldsymbol{x}, k)$ is ρ-admissible. Let a local feature value be bounded by v such that $\boldsymbol{x}_{(k,q)} \leq v$ for $q \in \{1, \ldots, d\}$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the random draw of the training set S, the generalization risk $R(f)$ is bounded by

$$R(f) \leq R_e(f) + \frac{2d\tau^2\rho^2v^2n^2}{m\lambda\alpha} + \Big(\frac{(4m-2)d\tau^2\rho^2v^2n^2}{m\lambda\alpha} + \gamma \Big) \sqrt{\frac{\ln \delta^{-1}}{2m}} \tag{9}$$

The proof is given in the full version of this work [16]. We call the term $\frac{2d\tau^2\rho^2v^2n^2}{m\lambda\alpha} + \Big(\frac{(4m-2)d\tau^2\rho^2v^2n^2}{m\lambda\alpha} + \gamma \Big) \sqrt{\frac{\ln \delta^{-1}}{2m}}$ in (9) as "overfit-bound", and reducing the overfit-bound is crucial for reducing the generalization risk. Most importantly, we can see from the overfit-bound that the structure regularization factor α is always staying together with the weight regularization factor λ, working together on reducing the overfit-bound. This indicates that the structure regularization is as important as the weight regularization for reducing the generalization risk.

Theorem 4 also indicates that too simple structures may overkill the overfit-bound but with a dominating empirical risk, and too complex structures may overkill the empirical risk but with a dominating overfit-bound. Thus, to achieve the best prediction accuracy, a balanced complexity of structures should be used for training the model.

2.3 Accelerating Convergence Rates in Training

We also analyze the impact on the convergence rate of online learning by applying structure regularization. Following prior work [10], our analysis is based on the stochastic gradient descent (SGD) with fixed learning rate. Let $g(\boldsymbol{w})$ be the structured prediction objective function and $\boldsymbol{w} \in \mathcal{W}$ is the weight vector. Recall that the SGD update with fixed learning rate η has a form like this:

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta \nabla g_{\boldsymbol{z}_t}(\boldsymbol{w}_t) \tag{10}$$

where $g_{\boldsymbol{z}}(\boldsymbol{w}_t)$ is the stochastic estimation of the objective function based on $\boldsymbol{z}$ which is randomly drawn from S. To state our convergence rate analysis results, we need several assumptions following (Nemirovski et al. 2009). We assume g is strongly convex with modulus c, that is, $\forall \boldsymbol{w}, \boldsymbol{w}' \in \mathcal{W}$,

$$g(\boldsymbol{w}') \geq g(\boldsymbol{w}) + (\boldsymbol{w}' - \boldsymbol{w})^T \nabla g(\boldsymbol{w}) + \frac{c}{2} ||\boldsymbol{w}' - \boldsymbol{w}||^2 \tag{11}$$

When g is strongly convex, there is a global optimum/minimizer $\boldsymbol{w}^*$. We also assume Lipschitz continuous differentiability of g with the constant q, that is, $\forall \boldsymbol{w}, \boldsymbol{w}' \in \mathcal{W}$,

$$||\nabla g(\boldsymbol{w}') - \nabla g(\boldsymbol{w})|| \leq q ||\boldsymbol{w}' - \boldsymbol{w}|| \tag{12}$$

It is also reasonable to assume that the norm of $\nabla g_{\boldsymbol{z}}(\boldsymbol{w})$ has almost surely positive correlation with the structure complexity of $\boldsymbol{z}$,[4] which can be quantified by a bound $\kappa \in \mathbb{R}^+$:

$$||\nabla g_{\boldsymbol{z}}(\boldsymbol{w})||_2 \leq \kappa |\boldsymbol{z}| \quad \text{almost surely for} \quad \forall \boldsymbol{w} \in \mathcal{W} \tag{13}$$

where $|\boldsymbol{z}|$ denotes the structure complexity of $\boldsymbol{z}$. Moreover, it is reasonable to assume

$$\eta c < 1 \tag{14}$$

because even the ordinary gradient descent methods will diverge if $\eta c > 1$. Then, we show that structure regularization can quadratically accelerate the SGD rates of convergence:

[4]Many models (e.g., CRFs) satisfy this assumption that the gradient of a larger sample is expected to have a larger norm.

Proposition 5 (Convergence rates vs. structure regularization) *With the aforementioned assumptions, let the SGD training have a learning rate defined as* $\eta = \frac{c\epsilon\beta\alpha^2}{q\kappa^2 n^2}$*, where* $\epsilon > 0$ *is a convergence tolerance value and* $\beta \in (0, 1]$*. Let* t *be a integer satisfying*

$$t \geq \frac{q\kappa^2 n^2 \log\left(qa_0/\epsilon\right)}{\epsilon\beta c^2\alpha^2} \tag{15}$$

where n *and* $\alpha \in [1, n]$ *is like before, and* a_0 *is the initial distance which depends on the initialization of the weights* $\boldsymbol{w}_0$ *and the minimizer* $\boldsymbol{w}^*$*, i.e.,* $a_0 = ||\boldsymbol{w}_0 - \boldsymbol{w}^*||^2$*. Then, after* t *updates of* $\boldsymbol{w}$ *it converges to* $\mathbb{E}[g(\boldsymbol{w}_t) - g(\boldsymbol{w}^*)] \leq \epsilon$*.*

The proof is given in the full version of this work [16]. As we can see, using structure regularization with the strength α can quadratically accelerate the convergence rate with a factor of α^2.

3 Experiments

Diversified Tasks. The natural language processing tasks include (1) part-of-speech tagging, (2) biomedical named entity recognition, and (3) Chinese word segmentation. The signal processing task is (4) sensor-based human activity recognition. The tasks (1) to (3) use boolean features and the task (4) adopts real-valued features. From tasks (1) to (4), the averaged structure complexity (number of observations) n is very different, with $n = 23.9, 26.5, 46.6, 67.9$, respectively. The dimension of tags $|\mathcal{Y}|$ is also diversified among tasks, with $|\mathcal{Y}|$ ranging from 5 to 45.

Part-of-Speech Tagging (POS-Tagging). Part-of-Speech (POS) tagging is an important and highly competitive task. We use the standard benchmark dataset in prior work [3], with 38,219 training samples and 5,462 test samples. Following prior work [22], we use features based on words and lexical patterns, with 393,741 raw features[5]. The evaluation metric is per-word accuracy.

Biomedical Named Entity Recognition (Bio-NER). This task is from the *BioNLP-2004 shared task* [22]. There are 17,484 training samples and 3,856 test samples. Following prior work [22], we use word pattern features and POS features, with 403,192 raw features in total. The evaluation metric is balanced F-score.

Word Segmentation (Word-Seg). We use the MSR data provided by *SIGHAN-2004 contest* [4]. There are 86,918 training samples and 3,985 test samples. The features are similar to [18], with 1,985,720 raw features in total. The evaluation metric is balanced F-score.

Sensor-based Human Activity Recognition (Act-Recog). This is a task based on real-valued sensor signals, with the data extracted from the Bao04 activity recognition dataset [17]. The features are similar to [17], with 1,228 raw features in total. There are 16,000 training samples and 4,000 test samples. The evaluation metric is accuracy.

We choose the CRFs [6] and structured perceptrons (Perc) [3], which are arguably the most popular probabilistic and non-probabilistic structured prediction models, respectively. The CRFs are trained using the SGD algorithm,[6] and the baseline method is the traditional weight regularization scheme (*WeightReg*), which adopts the most representative L_2 weight regularization, i.e., a Gaussian prior. We also tested sparsity emphasized regularization methods, including L_1 regularization and *Group Lasso* regularization [9]. However, although the feature sparsity is improved, we find in experiments that in most cases those sparsity emphasized regularization methods have lower accuracy than the L_2 regularization. For the structured perceptrons, the baseline *WeightAvg* is the popular implicit regularization technique based on parameter averaging, i.e., *averaged perceptron* [3].

The rich edge features [19, 18] are employed for all methods. All methods are based on the 1st-order Markov dependency. For *WeightReg*, the L_2 regularization strengths (i.e., $\lambda/2$ in Eq.(8)) are tuned among values $0.1, 0.5, 1, 2, 5$, and are determined on the development data (POS-Tagging) or simply via 4-fold cross validation on the training set (Bio-NER, Word-Seg, and Act-Recog). With this automatic tuning for *WeightReg*, we set $2, 5, 1$ and 5 for POS-Tagging, Bio-NER, Word-Seg, and Act-Recog tasks, respectively.

[5] Raw features are those observation features based only on $\boldsymbol{x}$, i.e., no combination with tag information.

[6] In theoretical analysis, following prior work we adopt the SGD with fixed learning rate, as described in Section 2.3. However, since the SGD with decaying learning rate is more commonly used in practice, in experiments we use the SGD with decaying learning rate.

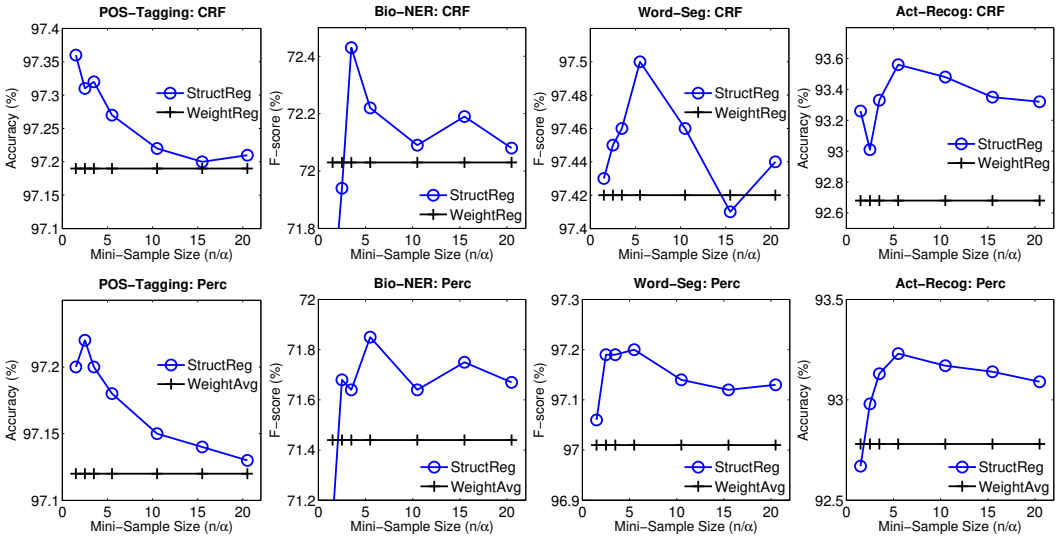

Figure 2: On the four tasks, comparing the structure regularization method (*StructReg*) with existing regularization methods in terms of accuracy/F-score. Row-1 shows the results on CRFs and Row-2 shows the results on structured perceptrons.

Table 1: Comparing our results with the benchmark systems on corresponding tasks.

	POS-Tagging (Acc%)	Bio-NER (F1%)	Word-Seg (F1%)
Benchmark system	97.33 (see [15])	72.28 (see [22])	97.19 (see [4])
Our results	**97.36**	**72.43**	**97.50**

3.1 Experimental Results

The experimental results in terms of accuracy/F-score are shown in Figure 2. For the CRF model, the training is convergent, and the results on the convergence state (decided by relative objective change with the threshold value of 0.0001) are shown. For the structured perceptron model, the training is typically not convergent, and the results on the 10'th iteration are shown. For stability of the curves, the results of the structured perceptrons are averaged over 10 repeated runs.

Since different samples have different size n in practice, we set α being a function of n, so that the generated mini-samples are with *fixed* size n' with $n' = n/\alpha$. Actually, n' is a probabilistic distribution because we adopt randomized decomposition. For example, if $n' = 5.5$, it means the mini-samples are a mixture of the ones with the size 5 and the ones with the size 6, and the mean of the size distribution is 5.5. In the figure, the curves are based on $n' = 1.5, 2.5, 3.5, 5.5, 10.5, 15.5, 20.5$.

As we can see, the results are quite consistent. It demonstrates that structure regularization leads to higher accuracies/F-scores compared with the existing baselines. We also conduct significance tests based on t-test. Since the t-test for F-score based tasks (Bio-NER and Word-Seg) may be unreliable[7], we only perform t-test for the accuracy-based tasks, i.e., POS-Tagging and Act-Recog. For POS-Tagging, the significance test suggests that the superiority of *StructReg* over *WeightReg* is very statistically significant, with $p < 0.01$. For Act-Recog, the significance tests suggest that both the *StructReg vs. WeightReg* difference and the *StructReg vs. WeightAvg* difference are extremely statistically significant, with $p < 0.0001$ in both cases. The experimental results support our theoretical analysis that structure regularization can further reduce the generalization risk over existing weight regularization techniques.

Our method outperforms the benchmark systems on the three important natural language processing tasks. The POS-Tagging task is a highly competitive task, with many methods proposed, and the best report (without using extra resources) until now is achieved by using a bidirectional learning model

[7]Indeed we can convert F-scores to accuracy scores for t-test, but in many cases this conversion is unreliable. For example, very different F-scores may correspond to similar accuracy scores.

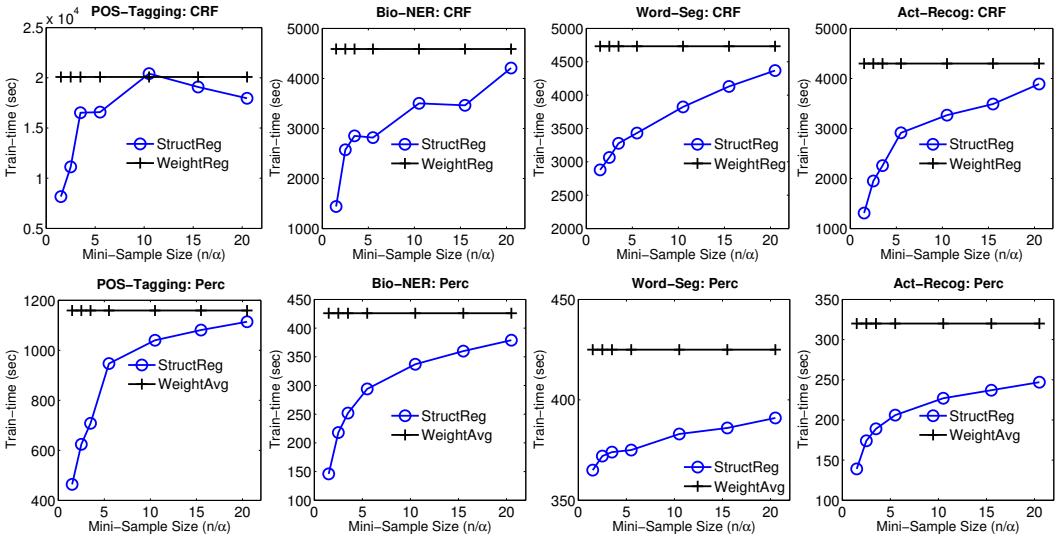

Figure 3: On the four tasks, comparing the structure regularization method (*StructReg*) with existing regularization methods in terms of wall-clock training time.

in [15],[8] with the accuracy 97.33%. Our simple method achieves better accuracy compared with all of those state-of-the-art systems. Furthermore, our method achieves as good scores as the benchmark systems on the Bio-NER and Word-Seg tasks. On the Bio-NER task, [22] achieves 72.28% based on lookahead learning and [24] achieves 72.65% based on reranking. On the Word-Seg task, [4] achieves 97.19% based on maximum entropy classification and our recent work [18] achieves 97.5% based on feature-frequency-adaptive online learning. The comparisons are summarized in Table 1.

Figure 3 shows experimental comparisons in terms of wall-clock training time. As we can see, the proposed method can substantially improve the training speed. The speedup is not only from the faster convergence rates, but also from the faster processing time on the structures, because it is more efficient to process the decomposed samples with simple structures.

4 Conclusions and Future Work

We proposed a structure regularization framework, which decomposes training samples into mini-samples with simpler structures, deriving a trained model with regularized structural complexity. Our theoretical analysis showed that this method can effectively reduce the generalization risk, and can also accelerate the convergence speed in training. The proposed method does not change the convexity of the objective function, and can be used together with any existing weight regularization methods. The proposed method and the theoretical results can fit general structures including linear chains, trees, and graphs. Experimental results demonstrated that our method achieved better results than state-of-the-art systems on several highly-competitive tasks, and at the same time with substantially faster training speed.

The structure decomposition of structure regularization can naturally used for parallel training, achieving parallel training among *mini-samples*. As future work, we will combine structure regularization with parallel training.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (No. 61300063), and Doctoral Fund of Ministry of Education of China (No. 20130001120004).

[8] See a collection of the systems at `http://aclweb.org/aclwiki/index.php?title=POS_Tagging_(State_of_the_art)`

References

[1] A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying. A spectral regularization framework for multi-task structure learning. In *Proceedings of NIPS'07*. MIT Press, 2007.

[2] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[3] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP'02*, pages 1–8, 2002.

[4] J. Gao, G. Andrew, M. Johnson, and K. Toutanova. A comparative study of parameter estimation methods for statistical natural language processing. In *Proceedings of ACL'07*, pages 824–831, 2007.

[5] J. Graça, K. Ganchev, B. Taskar, and F. Pereira. Posterior vs parameter sparsity in latent variable models. In *Proceedings of NIPS'09*, pages 664–672, 2009.

[6] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML'01*, pages 282–289, 2001.

[7] B. London, B. Huang, B. Taskar, and L. Getoor. Collective stability in structured prediction: Generalization from one example. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 828–836, 2013.

[8] B. London, B. Huang, B. Taskar, and L. Getoor. Pac-bayes generalization bounds for randomized structured prediction. In *NIPS Workshop on Perturbation, Optimization and Statistics*, 2013.

[9] A. F. T. Martins, N. A. Smith, M. A. T. Figueiredo, and P. M. Q. Aguiar. Structured sparsity in structured prediction. In *Proceedings of EMNLP'11*, pages 1500–1511, 2011.

[10] F. Niu, B. Recht, C. Re, and S. J. Wright. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS'11*, pages 693–701, 2011.

[11] A. Quattoni, X. Carreras, M. Collins, and T. Darrell. An efficient projection for l1,infinity regularization. In *Proceedings of ICML'09*, page 108, 2009.

[12] R. Samdani and D. Roth. Efficient decomposed learning for structured prediction. In *ICML'12*, 2012.

[13] M. W. Schmidt and K. P. Murphy. Convex structure learning in log-linear models: Beyond pairwise potentials. In *Proceedings of AISTATS'10*, volume 9 of *JMLR Proceedings*, pages 709–716, 2010.

[14] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Learnability and stability in the general learning setting. In *Proceedings of COLT'09*, 2009.

[15] L. Shen, G. Satta, and A. K. Joshi. Guided learning for bidirectional sequence classification. In *Proceedings of ACL'07*, 2007.

[16] X. Sun. Structure regularization for structured prediction: Theories and experiments. In *Technical report, arXiv*, 2014.

[17] X. Sun, H. Kashima, and N. Ueda. Large-scale personalized human activity recognition using online multitask learning. *IEEE Trans. Knowl. Data Eng.*, 25(11):2551–2563, 2013.

[18] X. Sun, W. Li, H. Wang, and Q. Lu. Feature-frequency-adaptive on-line training for fast and accurate natural language processing. *Computational Linguistics*, 40(3):563–586, 2014.

[19] X. Sun, H. Wang, and W. Li. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of ACL'12*, pages 253–262, 2012.

[20] C. A. Sutton and A. McCallum. Piecewise pseudolikelihood for efficient training of conditional random fields. In *ICML'07*, pages 863–870. ACM, 2007.

[21] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *NIPS'03*, 2003.

[22] Y. Tsuruoka, Y. Miyao, and J. Kazama. Learning with lookahead: Can history-based models rival globally optimized models? In *Conference on Computational Natural Language Learning*, 2011.

[23] H. Xue, S. Chen, and Q. Yang. Structural regularized support vector machine: A framework for structural large margin classifier. *IEEE Transactions on Neural Networks*, 22(4):573–587, 2011.

[24] K. Yoshida and J. Tsujii. Reranking for biomedical named-entity recognition. In *ACL Workshop on BioNLP*, pages 209–216, 2007.

[25] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

A Theoretical Analysis: Structure Complexity vs. Overfitting Risk

We first describe the settings for the theoretical analysis, and give the necessary definitions (Section A.1). We then introduce the proposed method with the proper annotations for clearance of the analysis (Section A.2). Finally, we give the theoretical results on analyzing the generalization risk regarding to the structure complexity based on stability (Section A.3). The general idea behind the theoretical analysis is that the overfitting risk increases with the complexity of the structure, because more complex structures are less stable in training. If some examples are taken out of the training set, the impact on the complex structure models is much severer compared to the simple structure models. The detailed relations among the factors are shown by the analysis.

A.1 Problem Settings of Theoretical Analysis

In this section, we give the preliminary definitions necessary for the analysis, including the learning algorithm, the data, and the cost functions, and especially the definition of structural complexity. We also describe the properties and the assumptions we make to facilitate the theoretical analysis.

A graph of observations (even with arbitrary structures) can be indexed and be denoted by an indexed sequence of observations $\boldsymbol{O} = \{o_1, \ldots, o_n\}$. We use the term *sample* to denote $\boldsymbol{O} = \{o_1, \ldots, o_n\}$. For example, in natural language processing, a sample may correspond to a sentence of n words with dependencies of linear chain structures (e.g., in part-of-speech tagging) or tree structures (e.g., in syntactic parsing). In signal processing, a sample may correspond to a sequence of n signals with dependencies of arbitrary structures. For simplicity in analysis, we assume all samples have n observations (thus n tags). In the analysis, we define structural complexity as the scope of the structural dependency. For example, a dependency scope of two tags is considered less complex than a dependency scope of three tags. In particular, the dependency scope of n tags is considered the full dependency scope which is of the highest structural complexity.

A sample is converted to an indexed sequence of feature vectors $\boldsymbol{x} = \{\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)}\}$, where $\boldsymbol{x}_{(k)} \in \mathcal{X}$ is of the dimension d and corresponds to the local features extracted from the position/index k. We can use an $n \times d$ matrix to represent $\boldsymbol{x} \in \mathcal{X}^n$. In other words, we use $\mathcal{X}$ to denote the input space on a position, so that $\boldsymbol{x}$ is sampled from $\mathcal{X}^n$. Let $\mathcal{Y}^n \subset \mathbb{R}^n$ be structured output space, so that the structured output $\boldsymbol{y}$ are sampled from $\mathcal{Y}^n$. Let $\mathcal{Z} = (\mathcal{X}^n, \mathcal{Y}^n)$ be a unified denotation of structured input and output space. Let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$, which is sampled from $\mathcal{Z}$, be a unified denotation of a $(\boldsymbol{x}, \boldsymbol{y})$ pair in the training data.

Suppose a training set is

$$S = \{\boldsymbol{z}_1 = (\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, \boldsymbol{z}_m = (\boldsymbol{x}_m, \boldsymbol{y}_m)\},$$

with size m, and the samples are drawn i.i.d. from a distribution D which is unknown. A learning algorithm is a function $G : \mathcal{Z}^m \mapsto \mathcal{F}$ with the function space $\mathcal{F} \subset \{\mathcal{X}^n \mapsto \mathcal{Y}^n\}$, i.e., G maps a training set S to a function $G_S : \mathcal{X}^n \mapsto \mathcal{Y}^n$. We suppose G is symmetric with respect to S, so that G is independent on the order of S.

Structural dependencies among tags are the major difference between structured prediction and non-structured classification. For the latter case, a local classification of g based on a position k can be expressed as $g(\boldsymbol{x}_{(k-a)}, \ldots, \boldsymbol{x}_{(k+a)})$, where the term $\{\boldsymbol{x}_{(k-a)}, \ldots, \boldsymbol{x}_{(k+a)}\}$ represents a local window. However, for structured prediction, a local classification on a position depends on the whole input $\boldsymbol{x} = \{\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)}\}$ rather than a local window, due to the nature of structural dependencies among tags (e.g., graphical models like CRFs). Thus, in structured prediction a local classification on k should be denoted as $g(\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)}, k)$. To simplify the notation, we define

$$g(\boldsymbol{x}, k) \triangleq g(\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(n)}, k)$$

Given a training set S of size m, we define $S^{\setminus i}$ as a modified training set, which removes the i'th training sample:

$$S^{\setminus i} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{i-1}, \boldsymbol{z}_{i+1}, \ldots, \boldsymbol{z}_m\},$$

and we define S^i as another modified training set, which replaces the i'th training sample with a new sample $\hat{\boldsymbol{z}}_i$ drawn from D:

$$S^i = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{i-1}, \hat{\boldsymbol{z}}_i, \boldsymbol{z}_{i+1}, \ldots, \boldsymbol{z}_m\},$$

We define the *point-wise cost function* $c : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}^+$ as $c[G_S(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}]$, which measures the cost on a position k by comparing $G_S(\boldsymbol{x}, k)$ and the gold-standard tag $\boldsymbol{y}_{(k)}$. We introduce the point-wise loss as

$$\ell(G_S, \boldsymbol{z}, k) \triangleq c[G_S(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}]$$

Then, we define the *sample-wise cost function* $C : \mathcal{Y}^n \times \mathcal{Y}^n \mapsto \mathbb{R}^+$, which is the cost function with respect to a whole sample. We introduce the sample-wise loss as

$$\mathcal{L}(G_S, \boldsymbol{z}) \triangleq C[G_S(\boldsymbol{x}), \boldsymbol{y}] = \sum_{k=1}^{n} \ell(G_S, \boldsymbol{z}, k) = \sum_{k=1}^{n} c[G_S(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}]$$

Given G and a training set S, what we are most interested in is the *generalization risk* in structured prediction (i.e., the expected average loss) [21, 8]:

$$R(G_S) = \mathbb{E}_{\boldsymbol{z}}\Big[\frac{\mathcal{L}(G_S, \boldsymbol{z})}{n}\Big]$$

Unless specifically indicated in the context, the probabilities and expectations over random variables, including $\mathbb{E}_{\boldsymbol{z}}(.)$, $\mathbb{E}_S(.)$, $\mathbb{P}_{\boldsymbol{z}}(.)$, and $\mathbb{P}_S(.)$, are based on the unknown distribution D.

Since the distribution D is unknown, we have to estimate $R(G_S)$ from S by using the *empirical risk*:

$$R_e(G_S) = \frac{1}{mn}\sum_{i=1}^{m}\mathcal{L}(G_S, \boldsymbol{z}_i) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{k=1}^{n}\ell(G_S, \boldsymbol{z}_i, k)$$

In what follows, sometimes we will use the simplified notations, R and R_e, to denote $R(G_S)$ and $R_e(G_S)$.

To state our theoretical results, we must describe several quantities and assumptions which are important in structured prediction. We follow some notations and assumptions on non-structured classification [2, 14]. We assume a simple real-valued structured prediction scheme such that the class predicted on position k of $\boldsymbol{x}$ is the sign of $G_S(\boldsymbol{x}, k) \in \mathcal{D}$. In practice, many popular structured prediction models have a real-valued cost function. Also, we assume the point-wise cost function c_τ is convex and τ-*smooth* such that $\forall y_1, y_2 \in \mathcal{D}, \forall y^* \in \mathcal{Y}$

$$|c_\tau(y_1, y^*) - c_\tau(y_2, y^*)| \leq \tau|y_1 - y_2| \tag{16}$$

While many structured learning models have convex objective function (e.g., CRFs), some other models have non-convex objective function (e.g., deep neural networks). It is well-known that the theoretical analysis on the non-convex cases are quite difficult. Our theoretical analysis is focused on the convex situations and hopefully it can provide some insight for the more difficult non-convex cases. In fact, we will conduct experiments on neural network models with non-convex objective functions, such as LSTM. Experimental results demonstrate that the proposed structural complexity regularization method also works in the non-convex situations, in spite of the difficulty of the theoretical analysis.

Then, τ-*smooth* versions of the loss and the cost function can be derived according to their prior definitions:

$$\mathcal{L}_\tau(G_S, \boldsymbol{z}) = C_\tau[G_S(\boldsymbol{x}), \boldsymbol{y}] = \sum_{k=1}^{n}\ell_\tau(G_S, \boldsymbol{z}, k) = \sum_{k=1}^{n}c_\tau[G_S(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}]$$

Also, we use a value ρ to quantify the bound of $|G_S(\boldsymbol{x}, k) - G_{S^{\setminus i}}(\boldsymbol{x}, k)|$ while changing a single sample (with size $n' \leq n$) in the training set with respect to the structured input $\boldsymbol{x}$. This ρ-*admissible* assumption can be formulated as $\forall k$,

$$|G_S(\boldsymbol{x}, k) - G_{S^{\setminus i}}(\boldsymbol{x}, k)| \leq \rho||G_S - G_{S^{\setminus i}}||_2 \cdot ||\boldsymbol{x}||_2 \tag{17}$$

where $\rho \in \mathbb{R}^+$ is a value related to the design of algorithm G.

A.2 Structural Complexity Regularization

Base on the problem settings, we give definitions for the common weight regularization and the proposed structural complexity regularization. In the definition, the proposed structural complexity regularization decomposes the dependency scope of the training samples into smaller localized dependency scopes. The smaller localized dependency scopes form mini-samples for the learning algorithms. It is assumed that the smaller localized dependency scopes are not overlapped. Hence, the analysis is for a simplified version of structural complexity regularization. We are aware that in implementation, the constraint can be hard to guarantee. From an empirical side, structural complexity works well without this constraint.

Most existing regularization techniques are proposed to regularize model weights/parameters, e.g., a representative regularizer is the Gaussian regularizer or so called L_2 regularizer. We call such regularization techniques as *weight regularization*.

Definition 6 (Weight regularization) *Let $N_\lambda : \mathcal{F} \mapsto \mathbb{R}^+$ be a weight regularization function on $\mathcal{F}$ with regularization strength λ, the structured classification based objective function with general weight regularization is as follows:*

$$R_\lambda(G_S) \triangleq R_e(G_S) + N_\lambda(G_S) \tag{18}$$

While weight regularization normalizes model weights, the proposed structural complexity regularization method normalizes the structural complexity of the training samples. Our analysis is based on the different *dependency scope* (i.e., the scope of the structural dependency), such that, for example, a tag depending on

two tags in context is considered to have less structural complexity than a tag depending on four tags in context. The structural complexity regularization is defined to make the *dependency scope* smaller. To simplify the analysis, we suppose a baseline case that a sample $\boldsymbol{z}$ has full dependency scope n, such that all tags in $\boldsymbol{z}$ have dependencies. Then, we introduce a factor α such that a sample $\boldsymbol{z}$ has localized dependency scope n/α. In this case, α represents the reduction magnitude of the dependency scope. To simplify the analysis without losing generality, we assume the localized dependency scopes do not overlap with each other. Since the dependency scope is localized and non-overlapping, we can split the original sample of the dependency scope n into α mini-samples of the dependency scope of n/α. What we want to show is that, the learning with small and non-overlapping dependency scope has less overfitting risk than the learning with large dependency scope. Real-world tasks may have an overlapping dependency scope. Hence, our theoretical analysis is for a simplified "essential" problem distilled from the real-world tasks.

In what follows, we also directly call the dependency scope of a sample as the *structure complexity* of the sample. Then, a simplified version of structural complexity regularization, specifically for our theoretical analysis, can be formally defined as follows:

Definition 7 (Simplified structural complexity regularization for analysis) *Let $N_\alpha : \mathcal{F} \mapsto \mathcal{F}$ be a structural complexity regularization function on $\mathcal{F}$ with regularization strength α with $1 \leq \alpha \leq n$, the structured classification based objective function with structural complexity regularization is as follows*[9]*:*

$$R_\alpha(G_S) \triangleq R_e[G_{N_\alpha(S)}] = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{\alpha}\mathcal{L}[G_{S'}, \boldsymbol{z}_{(i,j)}] = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{\alpha}\sum_{k=1}^{n/\alpha}\ell[G_{S'}, \boldsymbol{z}_{(i,j)}, k] \tag{19}$$

where $N_\alpha(\boldsymbol{z}_i)$ splits $\boldsymbol{z}_i$ into α mini-samples $\{\boldsymbol{z}_{(i,1)}, \ldots, \boldsymbol{z}_{(i,\alpha)}\}$, so that the mini-samples have a dependency scope of $n' = n/\alpha$. Thus, we get

$$S' = \{\underbrace{\boldsymbol{z}_{(1,1)}, z_{(1,2)}, \ldots, \boldsymbol{z}_{(1,\alpha)}}_{\alpha}, \ldots, \underbrace{\boldsymbol{z}_{(m,1)}, \boldsymbol{z}_{(m,2)}, \ldots, \boldsymbol{z}_{(m,\alpha)}}_{\alpha}\} \tag{20}$$

with $m\alpha$ mini-samples of expected structure complexity n/α. We can denote S' more compactly as $S' = \{\boldsymbol{z}'_1, \boldsymbol{z}'_2, \ldots, \boldsymbol{z}'_{m\alpha}\}$ and $R_\alpha(G_S)$ can be simplified as

$$R_\alpha(G_S) \triangleq \frac{1}{mn}\sum_{i=1}^{m\alpha}\mathcal{L}(G_{S'}, \boldsymbol{z}'_i) = \frac{1}{mn}\sum_{i=1}^{m\alpha}\sum_{k=1}^{n/\alpha}\ell[G_{S'}, \boldsymbol{z}'_i, k] \tag{21}$$

Note that, when the structural complexity regularization strength $\alpha = 1$, we have $S' = S$ and $R_\alpha = R_e$.

Now, we have given a formal definition of structural complexity regularization, by comparing it with the traditional weight regularization. Below, we show that the structural complexity regularization can improve the stability of learned models, and can finally reduce the overfitting risk of the learned models.

A.3 Stability of Structured Prediction

Because the generalization of a learning algorithm is positively correlated with the stability of the learning algorithm [2], to analyze the generalization of the proposed method, we instead examine the stability of the structured prediction. Here, stability describes the extent to which the resulting learning function changes, when a sample in the training set is removed. We prove that by decomposing the dependency scopes, i.e, by regularizing the structural complexity, the stability of the learning algorithm can be improved.

We first give the formal definitions of the stability with respect to the learning algorithm, i.e., function stability.

Definition 8 (Function stability) *A real-valued structured classification algorithm G has "function value based stability" ("function stability" for short) Δ if the following holds: $\forall \boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{Z}, \forall S \in \mathcal{Z}^m, \forall i \in \{1, \ldots, m\}, \forall k \in \{1, \ldots, n\}$,*

$$|G_S(\boldsymbol{x}, k) - G_{S^{\setminus i}}(\boldsymbol{x}, k)| \leq \Delta$$

The stability with respect to the cost function can be similarly defined.

Definition 9 (Loss stability) *A structured classification algorithm G has "uniform loss-based stability" ("loss stability" for short) Δ_l if the following holds: $\forall \boldsymbol{z} \in \mathcal{Z}, \forall S \in \mathcal{Z}^m, \forall i \in \{1, \ldots, m\}, \forall k \in \{1, \ldots, n\}$,*

$$|\ell(G_S, \boldsymbol{z}, k) - \ell(G_{S^{\setminus i}}, \boldsymbol{z}, k)| \leq \Delta_l$$

[9]The notation N is overloaded here. For clarity throughout, N with subscript λ refers to weight regularization function, and N with subscript α refers to structural complexity regularization function.

G has "sample-wise uniform loss-based stability" ("sample loss stability" for short) Δ_s with respect to the loss function $\mathcal{L}$ if the following holds: $\forall \boldsymbol{z} \in \mathcal{Z}, \forall S \in \mathcal{Z}^m, \forall i \in \{1, \ldots, m\}$,

$$|\mathcal{L}(G_S, \boldsymbol{z}) - \mathcal{L}(G_{S^{\setminus i}}, \boldsymbol{z})| \leq \Delta_s$$

It is clear that the upper bounds of loss stability and function stability are linearly correlated under the problem settings.

Lemma 10 (Loss stability vs. function stability) *If a real-valued structured classification algorithm G has function stability Δ with respect to loss function ℓ_τ, then G has loss stability*

$$\Delta_l \leq \tau \Delta$$

and sample loss stability

$$\Delta_s \leq n\tau\Delta.$$

The proof is provided in B.

Here, we show that lower structural complexity has lower bound of stability, and is more stable for the learning algorithm. The proposed method improves stability by regularizing the structural complexity of training samples.

Theorem 11 (Stability vs. structural complexity regularization) *With a training set S of size m, let the learning algorithm G have the minimizer f based on commonly used L_2 weight regularization:*

$$f = \operatorname*{argmin}_{g \in \mathcal{F}} R_{\alpha,\lambda}(g) = \operatorname*{argmin}_{g \in \mathcal{F}} \Big(\frac{1}{mn} \sum_{j=1}^{m\alpha} \mathcal{L}_\tau(g, \boldsymbol{z}'_j) + \frac{\lambda}{2} ||g||_2^2 \Big) \tag{22}$$

where α denotes the structural complexity regularization strength with $1 \leq \alpha \leq n$.

Also, we have

$$f^{\setminus i'} = \operatorname*{argmin}_{g \in \mathcal{F}} R^{\setminus i'}_{\alpha,\lambda}(g) = \operatorname*{argmin}_{g \in \mathcal{F}} \Big(\frac{1}{mn} \sum_{j \neq i'} \mathcal{L}_\tau(g, \boldsymbol{z}'_j) + \frac{\lambda}{2} ||g||_2^2 \Big) \tag{23}$$

where $j \neq i'$ means $j \in \{1, \ldots, i'-1, i'+1, \ldots, m\alpha\}$.[10] Assume $\mathcal{L}_\tau$ is convex and differentiable, and $f(\boldsymbol{x}, k)$ is ρ-admissible. Let a local feature value is bounded by v such that $\boldsymbol{x}_{(k,q)} \leq v$ for $q \in \{1, \ldots, d\}$.[11] Let Δ denote the function stability of f compared with $f^{\setminus i'}$ for $\forall \boldsymbol{z} \in \mathcal{Z}$ with $|\boldsymbol{z}| = n$. Then, Δ is bounded by

$$\Delta \leq \frac{d\tau\rho^2 v^2 n^2}{m\lambda\alpha^2}, \tag{24}$$

and the corresponding loss stability is bounded by

$$\Delta_l \leq \frac{d\tau^2\rho^2 v^2 n^2}{m\lambda\alpha^2},$$

and the corresponding sample loss stability is bounded by

$$\Delta_s \leq \frac{d\tau^2\rho^2 v^2 n^3}{m\lambda\alpha^2}.$$

The proof is given in B.

We can see that increasing the size of training set m results in linear improvement of Δ, and increasing the strength of structural complexity regularization α results in quadratic improvement of Δ.

The function stability Δ is based on comparing f and $f^{\setminus i'}$, i.e., the stability is based on removing a mini-sample. Moreover, we can extend the analysis to the function stability based on comparing f and $f^{\setminus i}$, i.e., the stability is based on removing a full-size sample.

[10] Note that, in some cases the notation i is ambiguous. For example, $f^{\setminus i}$ can either denote the removing of a sample in S or denote the removing of a mini-sample in S'. Thus, when the case is ambiguous, we use different index symbols for S and S', with i for indexing S and i' for indexing S', respectively.

[11] Recall that d is the dimension of local feature vectors defined in Section A.1.

Corollary 12 (Stability by removing a full sample) *With a training set S of size m, let the learning algorithm G have the minimizer f as defined before. Also, we have*

$$f^{\setminus i} = \operatorname*{argmin}_{g \in \mathcal{F}} R^{\setminus i}_{\alpha,\lambda}(g) = \operatorname*{argmin}_{g \in \mathcal{F}} \Big(\frac{1}{mn} \sum_{j \notin i} \mathcal{L}_\tau(g, \boldsymbol{z}'_j) + \frac{\lambda}{2} ||g||_2^2 \Big) \tag{25}$$

where $j \notin i$ means $j \in \{1, \ldots, (i-1)\alpha, i\alpha + 1, \ldots, m\alpha\}$, i.e., all the mini-samples derived from the sample $\boldsymbol{z}_i$ are removed. Assume $\mathcal{L}_\tau$ is convex and differentiable, and $f(\boldsymbol{x}, k)$ is ρ-admissible. Let a local feature value be bounded by v such that $\boldsymbol{x}_{(k,q)} \leq v$ for $q \in \{1, \ldots, d\}$. Let $\bar{\Delta}$ denote the function stability of f comparing with $f^{\setminus i}$ for $\forall \boldsymbol{z} \in \mathcal{Z}$ with $|\boldsymbol{z}| = n$. Then, $\bar{\Delta}$ is bounded by

$$\bar{\Delta} \leq \frac{d\tau\rho^2 v^2 n^2}{m\lambda\alpha} = \alpha \sup(\Delta), \tag{26}$$

where Δ is the function stability of f comparing with $f^{\setminus i'}$, and $\sup(\Delta) = \frac{d\tau\rho^2 v^2 n^2}{m\lambda\alpha^2}$, as described in Eq. (24). Similarly, we have

$$\bar{\Delta}_l \leq \frac{d\tau^2\rho^2 v^2 n^2}{m\lambda\alpha} = \alpha \sup(\Delta_l),$$

and

$$\bar{\Delta}_s \leq \frac{d\tau^2\rho^2 v^2 n^3}{m\lambda\alpha} = \alpha \sup(\Delta_s).$$

The proof is presented in B.

In the case that a full sample is removed, increasing the strength of structural complexity regularization α results in linear improvement of Δ.

A.4 Reduction of Generalization Risk

In this section, we formally describe the relation between the generalization and the stability, and summarize the relationship between the proposed method and the generalization. Finally, we draw our conclusions from the theoretical analysis.

Now, we analyze the relationship between the generalization and the stability.

Theorem 13 (Generalization vs. stability) *Let G be a real-valued structured classification algorithm with a point-wise loss function ℓ_τ such that $\forall k, 0 \leq \ell_\tau(G_S, \boldsymbol{z}, k) \leq \gamma$. Let f, Δ, and $\bar{\Delta}$ be defined before. Let $R(f)$ be the generalization risk of f based on the expected sample $\boldsymbol{z} \in \mathcal{Z}$ with size n, as defined before. Let $R_e(f)$ be the empirical risk of f based on S, as defined like before. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the random draw of the training set S, the generalization risk $R(f)$ is bounded by*

$$R(f) \leq R_e(f) + 2\tau\bar{\Delta} + \Big((4m - 2)\tau\bar{\Delta} + \gamma \Big) \sqrt{\frac{\ln \delta^{-1}}{2m}} \tag{27}$$

The proof is in B.

The upper bound of the generalization risk contains the loss stability, which is rewritten as the function stability. We can see that better stability leads to lower bound of the generalization risk.

By substituting the function stability with the formula we get from the structural complexity regularization, we get the relation between the generalization and the structural complexity regularization.

Theorem 14 (Generalization vs. structural complexity regularization) *Let the structured prediction objective function of G be penalized by structural complexity regularization with factor $\alpha \in [1, n]$ and L_2 weight regularization with factor λ. The penalized function has a minimizer f:*

$$f = \operatorname*{argmin}_{g \in \mathcal{F}} R_{\alpha,\lambda}(g) = \operatorname*{argmin}_{g \in \mathcal{F}} \Big(\frac{1}{mn} \sum_{j=1}^{m\alpha} \mathcal{L}_\tau(g, \boldsymbol{z}'_j) + \frac{\lambda}{2} ||g||_2^2 \Big) \tag{28}$$

Assume the point-wise loss ℓ_τ is convex and differentiable, and is bounded by $\ell_\tau(f, \boldsymbol{z}, k) \leq \gamma$. Assume $f(\boldsymbol{x}, k)$ is ρ-admissible. Let a local feature value be bounded by v such that $\boldsymbol{x}_{(k,q)} \leq v$ for $q \in \{1, \ldots, d\}$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the random draw of the training set S, the generalization risk $R(f)$ is bounded by

$$R(f) \leq R_e(f) + \frac{2d\tau^2\rho^2 v^2 n^2}{m\lambda\alpha} + \Big(\frac{(4m - 2)d\tau^2\rho^2 v^2 n^2}{m\lambda\alpha} + \gamma \Big) \sqrt{\frac{\ln \delta^{-1}}{2m}} \tag{29}$$

The proof is in B.

We call the term $\frac{2d\tau^2\rho^2v^2n^2}{m\lambda\alpha} + \Big(\frac{(4m-2)d\tau^2\rho^2v^2n^2}{m\lambda\alpha} + \gamma\Big)\sqrt{\frac{\ln\delta^{-1}}{2m}}$ in (9) as the "overfit-bound". Reducing the overfit-bound is crucial for reducing the generalization risk bound. Most importantly, we can see from the overfit-bound that the structural complexity regularization factor α always stays together with the weight regularization factor λ, working together to reduce the overfit-bound. This indicates that the structural complexity regularization is as important as the weight regularization for reducing the generalization risk for structured prediction.

Moreover, since τ, ρ, and v are typically small compared with other variables, especially m, (29) can be approximated as follows by ignoring the small terms:

$$R(f) \leq R_e(f) + O\Big(\frac{dn^2\sqrt{\ln\delta^{-1}}}{\lambda\alpha\sqrt{m}}\Big) \tag{30}$$

First, (30) suggests that structure complexity n can increase the overfit-bound on a magnitude of $O(n^2)$, and applying weight regularization can reduce the overfit-bound by $O(\lambda)$. Importantly, applying structural complexity regularization further (over weight regularization) can additionally reduce the overfit-bound by a magnitude of $O(\alpha)$. When $\alpha = 1$, which means "no structural complexity regularization", we have the worst overfit-bound. Also, (30) suggests that increasing the size of training set can reduce the overfit-bound on a square root level.

Theorem 14 also indicates that too simple structures may overkill the overfit-bound but with a dominating empirical risk, while too complex structures may overkill the empirical risk but with a dominating overfit-bound. Thus, to achieve the best prediction accuracy, a balanced complexity of structures should be used for training the model.

By regularizing the complex structure with the simple structure, a balance between the empirical risk and the overfitting risk can be achieved. In the proposed method, the model of the complex structure and the simple structure are both used in decoding. In essence, the decoding is based on the complex model, for the purpose of keeping the empirical risk down. The simple model is used to regularize the structure of the output, which means the structural complexity of the complex model is compromised. Therefore, the overfitting risk is reduced.

To summarize, the proposed method decomposes the dependency scopes, that is, regularizes the structural complexity. It leads to better stability of the model, which means the generalization risk is lower. Under the problem settings, increasing the regularization strength α can bring linear reduction of the overfit-bound. However, too simple structure may cause a dominating empirical risk. To achieve a balanced structural complexity, we could regularize the complex structure model with the simple structure model. The complex structure model has low empirical risk, while the simple structure model has low structural risk. The proposed method takes the advantages of both the simple structure model and the complex structure model. As a result, the overall overfitting risk can be reduced.

B Proof

Our analysis sometimes needs to use McDiarmid's inequality.

Theorem 15 (McDiarmid, 1989) *Let $S = \{q_1, \ldots, q_m\}$ be independent random variables taking values in the space Q^m. Moreover, let $g : Q^m \mapsto \mathbb{R}$ be a function of S that satisfies $\forall i, \forall S \in Q^m, \forall \hat{q}_i \in Q$,*

$$|g(S) - g(S^i)| \leq c_i.$$

Then $\forall \epsilon > 0$,

$$\mathbb{P}_S[g(S) - \mathbb{E}_S[g(S)] \geq \epsilon] \leq \exp\Big(\frac{-2\epsilon^2}{\sum_{i=1}^m c_i^2}\Big).$$

Lemma 16 (Symmetric learning) *For any symmetric (i.e., order-free) learning algorithm G, $\forall i \in \{1, \ldots, m\}$, we have*

$$\mathbb{E}_S[R(G_S) - R_e(G_S)] = \frac{1}{n}\mathbb{E}_{S,\hat{\boldsymbol{z}}_i}[\mathcal{L}(G_S, \hat{\boldsymbol{z}}_i) - \mathcal{L}(G_{S^i}, \hat{\boldsymbol{z}}_i)]$$

Proof

$$\begin{aligned}
\mathbb{E}_S[R(G_S) - R_e(G_S)] &= \frac{1}{n}\mathbb{E}_S\Big(\mathbb{E}_{\boldsymbol{z}}(\mathcal{L}(G_S, \boldsymbol{z})) - \frac{1}{m}\sum_{j=1}^{m}\mathcal{L}(G_S, \boldsymbol{z}_j)\Big) \\
&= \frac{1}{n}\Big(\mathbb{E}_{S,\hat{\boldsymbol{z}}_i}\big(\mathcal{L}(G_S, \hat{\boldsymbol{z}}_i)\big) - \frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_S\big(\mathcal{L}(G_S, \boldsymbol{z}_j)\big)\Big) \\
&= \frac{1}{n}\Big(\mathbb{E}_{S,\hat{\boldsymbol{z}}_i}\big(\mathcal{L}(G_S, \hat{\boldsymbol{z}}_i)\big) - \mathbb{E}_S\big(\mathcal{L}(G_S, \boldsymbol{z}_i)\big)\Big) \\
&= \frac{1}{n}\Big(\mathbb{E}_{S,\hat{\boldsymbol{z}}_i}\big(\mathcal{L}(G_S, \hat{\boldsymbol{z}}_i)\big) - \mathbb{E}_{S^i}\big(\mathcal{L}(G_{S^i}, \hat{\boldsymbol{z}}_i)\big)\Big) \\
&= \frac{1}{n}\mathbb{E}_{S,\hat{\boldsymbol{z}}_i}\big(\mathcal{L}(G_S, \hat{\boldsymbol{z}}_i) - \mathcal{L}(G_{S^i}, \hat{\boldsymbol{z}}_i)\big)
\end{aligned}$$

where the 3rd step is based on $\mathbb{E}_S\mathcal{L}(G_S, \boldsymbol{z}_i) = \mathbb{E}_S\mathcal{L}(G_S, \boldsymbol{z}_j)$ for $\forall \boldsymbol{z}_i \in S$ and $\forall \boldsymbol{z}_j \in S$, given that G is symmetric. □

B.1 Proofs

Proof of Lemma 10

According to (1), we have $\forall i, \forall S, \forall \boldsymbol{z}, \forall k$

$$\begin{aligned}
|\ell_\tau(G_S, \boldsymbol{z}, k) - \ell_\tau(G_{S^{\setminus i}}, \boldsymbol{z}, k)| &= |c_\tau[G_S(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}] - c_\tau[G_{S^{\setminus i}}(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}]| \\
&\leq \tau|G_S(\boldsymbol{x}, k) - G_{S^{\setminus i}}(\boldsymbol{x}, k)| \\
&\leq \tau\Delta
\end{aligned}$$

This gives the bound of loss stability.

Also, we have $\forall i, \forall S, \forall \boldsymbol{z}$

$$\begin{aligned}
|\mathcal{L}_\tau(G_S, \boldsymbol{z}) - \mathcal{L}_\tau(G_{S^{\setminus i}}, \boldsymbol{z})| &= \Big|\sum_{k=1}^{n} c_\tau[G_S(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}] - \sum_{k=1}^{n} c_\tau[G_{S^{\setminus i}}(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}]\Big| \\
&\leq \sum_{k=1}^{n}\Big|c_\tau[G_S(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}] - c_\tau[G_{S^{\setminus i}}(\boldsymbol{x}, k), \boldsymbol{y}_{(k)}]\Big| \\
&\leq \tau\sum_{k=1}^{n}|G_S(\boldsymbol{x}, k) - G_{S^{\setminus i}}(\boldsymbol{x}, k)| \\
&\leq n\tau\Delta
\end{aligned}$$

This derives the bound of sample loss stability. □

Proof of Theorem 11

When a convex and differentiable function g has a minimum f in space $\mathcal{F}$, its Bregman divergence has the following property for $\forall f' \in \mathcal{F}$:

$$d_g(f', f) = g(f') - g(f)$$

With this property, we have

$$\begin{aligned}
d_{R_{\alpha,\lambda}}(f^{\setminus i'}, f) + d_{R^{\setminus i'}_{\alpha,\lambda}}(f, f^{\setminus i'}) &= R_{\alpha,\lambda}(f^{\setminus i'}) - R_{\alpha,\lambda}(f) + R^{\setminus i'}_{\alpha,\lambda}(f) - R^{\setminus i'}_{\alpha,\lambda}(f^{\setminus i'}) \\
&= \big(R_{\alpha,\lambda}(f^{\setminus i'}) - R^{\setminus i'}_{\alpha,\lambda}(f^{\setminus i'})\big) - \big(R_{\alpha,\lambda}(f) - R^{\setminus i'}_{\alpha,\lambda}(f)\big) \qquad (31) \\
&= \frac{1}{mn}\mathcal{L}_\tau(f^{\setminus i'}, \boldsymbol{z}'_{i'}) - \frac{1}{mn}\mathcal{L}_\tau(f, \boldsymbol{z}'_{i'})
\end{aligned}$$

Then, based on the property of Bregman divergence that $d_{g+g'} = d_g + d_{g'}$, we have

$$\begin{aligned}
d_{N_\lambda}(f, f^{\setminus i'}) + d_{N_\lambda}(f^{\setminus i'}, f) &= d_{(R^{\setminus i'}_{\alpha,\lambda} - R^{\setminus i'}_{\alpha})}(f, f^{\setminus i'}) + d_{(R_{\alpha,\lambda} - R_\alpha)}(f^{\setminus i'}, f) \\
&= d_{R_{\alpha,\lambda}}(f^{\setminus i'}, f) + d_{R^{\setminus i'}_{\alpha,\lambda}}(f, f^{\setminus i'}) - d_{R_\alpha}(f^{\setminus i'}, f) - d_{R^{\setminus i'}_{\alpha}}(f, f^{\setminus i'}) \\
&\text{(based on non-negativity of Bregman divergence)} \\
&\leq d_{R_{\alpha,\lambda}}(f^{\setminus i'}, f) + d_{R^{\setminus i'}_{\alpha,\lambda}}(f, f^{\setminus i'}) \\
&\text{(using (31))} \\
&= \frac{1}{mn}\big(\mathcal{L}_\tau(f^{\setminus i'}, \boldsymbol{z}'_{i'}) - \mathcal{L}_\tau(f, \boldsymbol{z}'_{i'})\big) \\
&= \frac{1}{mn}\sum_{k=1}^{n/\alpha}\big(\ell_\tau(f^{\setminus i'}, \boldsymbol{z}'_{i'}, k) - \ell_\tau(f, \boldsymbol{z}'_{i'}, k)\big) \\
&\leq \frac{1}{mn}\sum_{k=1}^{n/\alpha}\Big|c_\tau\Big(f^{\setminus i'}(\boldsymbol{x}'_{i'}, k), \boldsymbol{y}'_{i'(k)}\Big) - c_\tau\Big(f(\boldsymbol{x}'_{i'}, k), \boldsymbol{y}'_{i'(k)}\Big)\Big| \\
&\leq \frac{\tau}{mn}\sum_{k=1}^{n/\alpha}\Big|f^{\setminus i'}(\boldsymbol{x}'_{i'}, k) - f(\boldsymbol{x}'_{i'}, k)\Big| \\
&\text{(using (17))} \\
&\leq \frac{\rho\tau}{m\alpha}||f - f^{\setminus i'}||_2 \cdot ||\boldsymbol{x}'_{i'}||_2
\end{aligned} \tag{32}$$

Moreover, $N_\lambda(g) = \frac{\lambda}{2}||g||_2^2 = \frac{\lambda}{2}\langle g, g\rangle$ is a convex function and its Bregman divergence satisfies:

$$\begin{aligned}
d_{N_\lambda}(g, g') &= \frac{\lambda}{2}\big(\langle g, g\rangle - \langle g', g'\rangle - \langle 2g', g - g'\rangle\big) \\
&= \frac{\lambda}{2}||g - g'||_2^2
\end{aligned} \tag{33}$$

Combining (32) and (33) gives

$$\lambda||f - f^{\setminus i'}||_2^2 \leq \frac{\rho\tau}{m\alpha}||f - f^{\setminus i'}||_2 \cdot ||\boldsymbol{x}'_{i'}||_2 \tag{34}$$

which further gives

$$||f - f^{\setminus i'}||_2 \leq \frac{\rho\tau}{m\lambda\alpha}||\boldsymbol{x}'_{i'}||_2 \tag{35}$$

Given ρ-admissibility, we derive the bound of function stability $\Delta(f)$ based on sample $\boldsymbol{z}$ with size n. We have $\forall \boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}), \forall k$,

$$\begin{aligned}
|f(\boldsymbol{x}, k) - f^{\setminus i'}(\boldsymbol{x}, k)| &\leq \rho||f - f^{\setminus i'}||_2 \cdot ||\boldsymbol{x}||_2 \\
&\text{(using (35))} \\
&\leq \frac{\tau\rho^2}{m\lambda\alpha}||\boldsymbol{x}'_{i'}||_2 \cdot ||\boldsymbol{x}||_2
\end{aligned} \tag{36}$$

With the feature dimension d and $\boldsymbol{x}_{(k,q)} \leq v$ for $q \in \{1, \ldots, d\}$, we have

$$\begin{aligned}
||\boldsymbol{x}||_2 &= ||\sum_{k=1}^{n}\boldsymbol{x}_{(k)}||_2 \\
&\leq ||\langle\underbrace{nv, \ldots, nv}_{d}\rangle||_2 \\
&= \sqrt{dn^2v^2} \\
&= nv\sqrt{d}
\end{aligned} \tag{37}$$

Similarly, we have $||\boldsymbol{x}'_{i'}||_2 \leq \frac{nv\sqrt{d}}{\alpha}$ because $\boldsymbol{x}'_{i'}$ is with the size n/α.

Inserting the bounds of $||\boldsymbol{x}||_2$ and $||\boldsymbol{x}'_{i'}||_2$ into (36), it goes to

$$|f(\boldsymbol{x}, k) - f^{\setminus i'}(\boldsymbol{x}, k)| \leq \frac{d\tau\rho^2v^2n^2}{m\lambda\alpha^2} \tag{38}$$

which gives (24). Further, using Lemma 10 derives the loss stability bound of $\frac{d\tau^2\rho^2v^2n^2}{m\lambda\alpha^2}$, and the sample loss stability bound of $\frac{d\tau^2\rho^2v^2n^3}{m\lambda\alpha^2}$ on the minimizer f. □

Proof of Corollary 12

The proof is similar to the proof of Theorem 11. First, we have

$$\begin{aligned} d_{R_{\alpha,\lambda}}(f^{\setminus i}, f) + d_{R^{\setminus i}_{\alpha,\lambda}}(f, f^{\setminus i}) &= R_{\alpha,\lambda}(f^{\setminus i}) - R_{\alpha,\lambda}(f) + R^{\setminus i}_{\alpha,\lambda}(f) - R^{\setminus i}_{\alpha,\lambda}(f^{\setminus i}) \\ &= \big(R_{\alpha,\lambda}(f^{\setminus i}) - R^{\setminus i}_{\alpha,\lambda}(f^{\setminus i})\big) - \big(R_{\alpha,\lambda}(f) - R^{\setminus i}_{\alpha,\lambda}(f)\big) \\ &= \frac{1}{mn}\sum_{j=1}^{\alpha}\mathcal{L}_\tau(f^{\setminus i}, \boldsymbol{z}_{(i,j)}) - \frac{1}{mn}\sum_{j=1}^{\alpha}\mathcal{L}_\tau(f, \boldsymbol{z}_{(i,j)}) \end{aligned} \tag{39}$$

Then, we have

$$\begin{aligned} d_{N_\lambda}(f, f^{\setminus i}) + d_{N_\lambda}(f^{\setminus i}, f) &= d_{(R^{\setminus i}_{\alpha,\lambda} - R^{\setminus i}_{\alpha})}(f, f^{\setminus i}) + d_{(R_{\alpha,\lambda} - R_\alpha)}(f^{\setminus i}, f) \\ &= d_{R_{\alpha,\lambda}}(f^{\setminus i}, f) + d_{R^{\setminus i}_{\alpha,\lambda}}(f, f^{\setminus i}) - d_{R_\alpha}(f^{\setminus i}, f) - d_{R^{\setminus i}_{\alpha}}(f, f^{\setminus i}) \\ &\text{(based on non-negativity of Bregman divergence)} \\ &\leq d_{R_{\alpha,\lambda}}(f^{\setminus i}, f) + d_{R^{\setminus i}_{\alpha,\lambda}}(f, f^{\setminus i}) \\ &\text{(using (39))} \\ &= \frac{1}{mn}\sum_{j=1}^{\alpha}\mathcal{L}_\tau(f^{\setminus i}, \boldsymbol{z}_{(i,j)}) - \frac{1}{mn}\sum_{j=1}^{\alpha}\mathcal{L}_\tau(f, \boldsymbol{z}_{(i,j)}) \\ &= \frac{1}{mn}\sum_{j=1}^{\alpha}\left(\sum_{k=1}^{n/\alpha}\ell_\tau(f^{\setminus i}, \boldsymbol{z}_{(i,j)}, k) - \sum_{k=1}^{n/\alpha}\ell_\tau(f, \boldsymbol{z}_{(i,j)}, k)\right) \\ &\leq \frac{1}{mn}\sum_{j=1}^{\alpha}\sum_{k=1}^{n/\alpha}\Big|\ell_\tau(f^{\setminus i}, \boldsymbol{z}_{(i,j)}, k) - \ell_\tau(f, \boldsymbol{z}_{(i,j)}, k)\Big| \\ &\leq \frac{\tau}{mn}\sum_{j=1}^{\alpha}\sum_{k=1}^{n/\alpha}\Big|f^{\setminus i}(\boldsymbol{x}_{(i,j)}, k) - f(\boldsymbol{x}_{(i,j)}, k)\Big| \\ &\text{(using (2), and define } ||\boldsymbol{x}_{(i,max)}||_2 = \max_{\forall j}||\boldsymbol{x}_{(i,j)}||)_2) \\ &\leq \frac{\rho\tau}{m}||f - f^{\setminus i}||_2 \cdot ||\boldsymbol{x}_{(i,max)}||_2 \end{aligned} \tag{40}$$

This gives

$$\lambda||f - f^{\setminus i}||_2^2 \leq \frac{\rho\tau}{m}||f - f^{\setminus i}||_2 \cdot ||\boldsymbol{x}_{(i,max)}||_2 \tag{41}$$

and thus

$$||f - f^{\setminus i}||_2 \leq \frac{\rho\tau}{m\lambda}||\boldsymbol{x}_{(i,max)}||_2 \tag{42}$$

Then, we derive the bound of function stability $\Delta(f)$ based on sample $\boldsymbol{z}$ with size n, and based on $\setminus i$ rather than $\setminus i'$. We have $\forall \boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}), \forall k$,

$$\begin{aligned} |f(\boldsymbol{x}, k) - f^{\setminus i}(\boldsymbol{x}, k)| &\leq \rho||f - f^{\setminus i}||_2 \cdot ||\boldsymbol{x}||_2 \\ &\text{(using (42))} \\ &\leq \frac{\tau\rho^2}{m\lambda}||\boldsymbol{x}_{(i,max)}||_2 \cdot ||\boldsymbol{x}||_2 \\ &\leq \frac{\tau\rho^2}{m\lambda} \cdot \frac{nv\sqrt{d}}{\alpha} \cdot nv\sqrt{d} \\ &= \frac{d\tau\rho^2v^2n^2}{m\lambda\alpha} \\ &\text{(using (24))} \\ &= \alpha\sup(\Delta) \end{aligned} \tag{43}$$

□

Proof of Theorem 13

Let $f^{\setminus i}$ be defined like before. Similar to the definition of $f^{\setminus i}$ based on *removing* a sample from S, we define f^i based on *replacing* a sample from S. Let $R(f)^{\setminus i}$ denote $[R(f)]^{\setminus i} = R^{\setminus i}(f^{\setminus i})$.

First, we derive a bound for $|R(f) - R^{\setminus i}(f)|$:

$$\begin{aligned}
|R(f) - R(f)^{\setminus i}| &= \frac{1}{n}|\mathbb{E}_{\boldsymbol{z}}\mathcal{L}_\tau(f, \boldsymbol{z}) - \mathbb{E}_{\boldsymbol{z}}\mathcal{L}_\tau(f^{\setminus i}, \boldsymbol{z})| \\
&= \frac{1}{n}|\mathbb{E}_{\boldsymbol{z}}\sum_{k=1}^{n}\ell_\tau(f, \boldsymbol{z}, k) - \mathbb{E}_{\boldsymbol{z}}\sum_{k=1}^{n}\ell_\tau(f^{\setminus i}, \boldsymbol{z}, k)| \\
&\leq \frac{1}{n}\mathbb{E}_{\boldsymbol{z}}|\sum_{k=1}^{n}\ell_\tau(f, \boldsymbol{z}, k) - \sum_{k=1}^{n}\ell_\tau(f^{\setminus i}, \boldsymbol{z}, k)| \\
&\leq \frac{1}{n}\mathbb{E}_{\boldsymbol{z}}\sum_{k=1}^{n}|\ell_\tau(f, \boldsymbol{z}, k) - \ell_\tau(f^{\setminus i}, \boldsymbol{z}, k)| \\
&\text{(based on Lemma 10 and the definition of } \bar{\Delta}) \\
&\leq \tau\bar{\Delta}
\end{aligned} \tag{44}$$

Then, we derive a bound for $|R(f) - R(f)^i|$:

$$\begin{aligned}
|R(f) - R(f)^i| &= |R(f) - R(f)^{\setminus i} + R(f)^{\setminus i} - R(f)^i| \\
&\leq |R(f) - R(f)^{\setminus i}| + |R(f)^{\setminus i} - R(f)^i| \\
&\text{(based on (44))} \\
&\leq \tau\bar{\Delta} + \tau\bar{\Delta} \\
&= 2\tau\bar{\Delta}
\end{aligned}$$

Moreover, we derive a bound for $|R_e(f) - R_e(f)^i|$. Let $\hat{\boldsymbol{z}}_i$ denote the full-size sample (with size n and indexed by i) which replaces the sample $\boldsymbol{z}_i$, it goes to:

$$\begin{aligned}
&|R_e(f) - R_e(f)^i| = \Big|\frac{1}{mn}\sum_{j=1}^{m}\mathcal{L}_\tau(f, \boldsymbol{z}_j) - \frac{1}{mn}\sum_{j\neq i}\mathcal{L}_\tau(f^i, \boldsymbol{z}_j) - \frac{1}{mn}\mathcal{L}_\tau(f^i, \hat{\boldsymbol{z}}_i)\Big| \\
&\leq \frac{1}{mn}\sum_{j\neq i}|\mathcal{L}_\tau(f, \boldsymbol{z}_j) - \mathcal{L}_\tau(f^i, \boldsymbol{z}_j)| + \frac{1}{mn}|\mathcal{L}_\tau(f, \boldsymbol{z}_i) - \mathcal{L}_\tau(f^i, \hat{\boldsymbol{z}}_i)| \\
&\leq \frac{1}{mn}\sum_{j\neq i}|\mathcal{L}_\tau(f, \boldsymbol{z}_j) - \mathcal{L}_\tau(f^i, \boldsymbol{z}_j)| + \frac{1}{mn}\sum_{k=1}^{n}|\ell_\tau(f, \boldsymbol{z}_i, k) - \ell_\tau(f^i, \hat{\boldsymbol{z}}_i, k)| \\
&\text{(based on } 0 \leq \ell_\tau(G_S, \boldsymbol{z}, k) \leq \gamma) \\
&\leq \frac{1}{mn}\sum_{j\neq i}|\mathcal{L}_\tau(f, \boldsymbol{z}_j) - \mathcal{L}_\tau(f^i, \boldsymbol{z}_j)| + \frac{\gamma}{m} \\
&\leq \frac{1}{mn}\sum_{j\neq i}\Big(|\mathcal{L}_\tau(f, \boldsymbol{z}_j) - \mathcal{L}_\tau(f^{\setminus i}, \boldsymbol{z}_j)| + |\mathcal{L}_\tau(f^{\setminus i}, \boldsymbol{z}_j) - \mathcal{L}_\tau(f^i, \boldsymbol{z}_j)|\Big) + \frac{\gamma}{m} \\
&\text{(based on Lemma 10, and } \Delta(f^i, f^{\setminus i}) = \Delta(f, f^{\setminus i}) \text{ from the definition of stability)} \\
&\leq \frac{1}{mn}\sum_{j\neq i}\Big(n\tau\bar{\Delta} + n\tau\bar{\Delta}\Big) + \frac{\gamma}{m} \\
&= \frac{2(m-1)\tau\bar{\Delta} + \gamma}{m}
\end{aligned} \tag{45}$$

Based on the bounds of $|R(f) - R(f)^i|$ and $|R_e(f) - R_e(f)^i|$, we show that $R(f) - R_e(f)$ satisfies the conditions of *McDiarmid Inequality* (Theorem 15) with $c_i = \frac{(4m-2)\tau\bar{\Delta}+\gamma}{m}$:

$$\begin{aligned}
|[R(f) - R_e(f)] - [R(f) - R_e(f)]^i| &= |[R(f) - R(f)^i] - [R_e(f) - R_e(f)^i]| \\
&\leq |R(f) - R(f)^i| + |R_e(f) - R_e(f)^i| \\
&\leq 2\tau\bar{\Delta} + \frac{2(m-1)\tau\bar{\Delta}+\gamma}{m} \\
&= \frac{(4m-2)\tau\bar{\Delta}+\gamma}{m}
\end{aligned} \tag{46}$$

Also, following the proof of Lemma 16, we can get a bound for $\mathbb{E}_S[R(f) - R_e(f)]$:

$$\begin{aligned}
\mathbb{E}_S[R(f) - R_e(f)] &= \frac{1}{n}\mathbb{E}_S\Big(\mathbb{E}_{\boldsymbol{z}}(\mathcal{L}(f,\boldsymbol{z})) - \frac{1}{m}\sum_{j=1}^{m}\mathcal{L}(f,\boldsymbol{z}_j)\Big) \\
&= \frac{1}{n}\Big(\mathbb{E}_{S,\hat{\boldsymbol{z}}_i}\big(\mathcal{L}(f,\hat{\boldsymbol{z}}_i)\big) - \frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_S\big(\mathcal{L}(f,\boldsymbol{z}_j)\big)\Big) \\
&= \frac{1}{n}\Big(\mathbb{E}_{S,\hat{\boldsymbol{z}}_i}\big(\mathcal{L}(f,\hat{\boldsymbol{z}}_i)\big) - \mathbb{E}_S\big(\mathcal{L}(f,\boldsymbol{z}_i)\big)\Big) \\
&= \frac{1}{n}\Big(\mathbb{E}_{S,\hat{\boldsymbol{z}}_i}\big(\mathcal{L}(f,\hat{\boldsymbol{z}}_i)\big) - \mathbb{E}_{S^i}\big(\mathcal{L}(f^i,\hat{\boldsymbol{z}}_i)\big)\Big) \\
&= \frac{1}{n}\mathbb{E}_{S,\hat{\boldsymbol{z}}_i}\big(\mathcal{L}(f,\hat{\boldsymbol{z}}_i) - \mathcal{L}(f^i,\hat{\boldsymbol{z}}_i)\big) \\
&\leq \frac{1}{n}\mathbb{E}_{S,\hat{\boldsymbol{z}}_i}|\mathcal{L}(f,\hat{\boldsymbol{z}}_i) - \mathcal{L}(f^i,\hat{\boldsymbol{z}}_i)| \\
&\leq \frac{1}{n}\mathbb{E}_{S,\hat{\boldsymbol{z}}_i}|\mathcal{L}(f,\hat{\boldsymbol{z}}_i) - \mathcal{L}(f^{\setminus i},\hat{\boldsymbol{z}}_i)| + \frac{1}{n}\mathbb{E}_{S,\hat{\boldsymbol{z}}_i}|\mathcal{L}(f^{\setminus i},\hat{\boldsymbol{z}}_i) - \mathcal{L}(f^i,\hat{\boldsymbol{z}}_i)| \\
&\text{(based on Lemma 10 and the } \bar{\Delta} \text{ defined in (26))} \\
&\leq \tau\bar{\Delta} + \tau\bar{\Delta} \\
&= 2\tau\bar{\Delta}
\end{aligned} \tag{47}$$

Now, we can apply *McDiarmid Inequality* (Theorem 15):

$$\mathbb{P}_S\Big([R(f) - R_e(f)] - \mathbb{E}_S[R(f) - R_e(f)] \geq \epsilon\Big) \leq \exp\Big(\frac{-2\epsilon^2}{\sum_{i=1}^{m} c_i^2}\Big) \tag{48}$$

Based on (46) and (47), it goes to

$$\mathbb{P}_S\Big(R(f) - R_e(f) \geq 2\tau\bar{\Delta} + \epsilon\Big) \leq \exp\left(\frac{-2m\epsilon^2}{\big((4m-2)\tau\bar{\Delta}+\gamma\big)^2}\right) \tag{49}$$

Let $\delta = \exp\Big(\frac{-2m\epsilon^2}{\big((4m-2)\tau\bar{\Delta}+\gamma\big)^2}\Big)$, we have

$$\epsilon = \Big((4m-2)\tau\bar{\Delta}+\gamma\Big)\sqrt{\frac{\ln\delta^{-1}}{2m}} \tag{50}$$

Based on (49) and (50), there is a probability no more than δ such that

$$\begin{aligned}
R(f) - R_e(f) &\geq 2\tau\bar{\Delta} + \epsilon \\
&= 2\tau\bar{\Delta} + \Big((4m-2)\tau\bar{\Delta}+\gamma\Big)\sqrt{\frac{\ln\delta^{-1}}{2m}}
\end{aligned} \tag{51}$$

Then, there is a probability at least $1-\delta$ such that

$$R(f) \leq R_e(f) + 2\tau\bar{\Delta} + \Big((4m-2)\tau\bar{\Delta}+\gamma\Big)\sqrt{\frac{\ln\delta^{-1}}{2m}}$$

which gives (27). □

Proof of Theorem 14

According to (26), we have $\bar{\Delta} \leq \frac{d\tau\rho^2 v^2 n^2}{m\lambda\alpha}$.

Inserting this bound into (27) gives (29). □