# Fine-grained Generalization Analysis of Structured Output Prediction

**Waleed Mustafa**[1] , **Yunwen Lei**[2] , **Antoine Ledent**[1] and **Marius Kloft**[1]
[1]TU Kaiserslautern
[2] University of Birmingham
mustafa@cs.uni-kl.de, y.lei@bham.ac.uk, {ledent, kloft}@cs.uni-kl.de

## Abstract

In machine learning we often encounter structured output prediction problems (SOPPs), i.e. problems where the output space admits a rich internal structure. Application domains where SOPPs naturally occur include natural language processing, speech recognition, and computer vision. Typical SOPPs have an extremely large label set, which grows exponentially as a function of the size of the output. Existing generalization analysis implies generalization bounds with at least a square-root dependency on the cardinality $d$ of the label set, which can be vacuous in practice. In this paper, we significantly improve the state of the art by developing novel high-probability bounds with a logarithmic dependency on $d$. Moreover, we leverage the lens of algorithmic stability to develop generalization bounds in expectation without any dependency on $d$. Our results therefore build a solid theoretical foundation for learning in large-scale SOPPs. Furthermore, we extend our results to learning with weakly dependent data.

## 1 Introduction

Structured output prediction (SOP) refers to a broad class of machine learning problems with a rich structure in the output space. For instance, the output may be a sequence of tags in part-of-speech (POS) tagging, a sentence in machine translation, or a grid of segmentation labels in image segmentation.

A distinguishing property of these tasks is that the loss function admits a decomposition along the output structures. For instance, if the output is a sequence of partial labels, the loss function could be the Hamming distance. The output structure makes those problems substantially different, both algorithmically and theoretically, from well-studied machine-learning methods such as binary classification. Algorithms specifically targeted at SOPPs have been put forward in [Lafferty *et al.*, 2001; Ciliberto *et al.*, 2016; Taskar *et al.*, 2003; Tsochantaridis *et al.*, 2005; Vinyals *et al.*, 2015; Lucchi *et al.*, 2013; Chen *et al.*, 2017], to mention but a few.

Whilst the subject of SOP is well explored from a practical point of view, existing theoretical analyses have several limitations. For instance, the results in [Taskar *et al.*, 2003; Collins, 2001] apply only to specific factor graphs and bound errors measured only by the Hamming loss, while other losses such as *edit distance* and *BLUE scores* are more natural in many applications. [McAllester, 2007] introduced guarantees that apply to general losses but only to randomized linear algorithms and admit only a square-root dependence on the size of substructures. In [Cortes *et al.*, 2016], the authors introduced general bounds that apply to general factor graphs and general losses from the viewpoint of function class capacity. However, the associated bounds exhibit a square-root dependence on the number $d$ of categories a subset of substructures can take, which can become vacuous when applied to extreme multi-class contexts [Lei *et al.*, 2019] or models that assume a large dependence between the substructures.

In this paper, we aim to advance the state of the art in the theoretical foundation of SOP by developing generalization bounds applicable to large-scale problems with millions of labels. Our contributions are as follows.

1. We apply the celebrated technique of Rademacher complexity to develop high-probability generalization bounds with a *log* dependency on the size of the label set. This substantially improves the existing state of the art, which comes with at least a square-root dependency. We achieve this improvement by using covering numbers measured by the $\ell_\infty$-norm, which can exploit the Lipschitz continuity of loss functions with respect to (w.r.t.) the $\ell_\infty$-norm. For comparison, the existing complexity analysis uses the Lipschitz continuity w.r.t. the $\ell_2$-norm [Cortes *et al.*, 2016], which does not match the regularity of loss functions in structured output prediction and thus leads to suboptimal bounds.
2. We leverage the framework of algorithmic stability to further remove the log dependency for generalization bounds in expectation. We consider two popular methods for structured output prediction: stochastic gradient descent (SGD) and regularized risk minimization (RRM). We adapt the existing stability analysis in a way to exploit the Lipschitz continuity w.r.t. the $\ell_\infty$-norm of loss functions in SOP.
3. We extend our discussion to learning with weakly dependent training examples, which are widespread in SOPPs. For example, in natural language processing (NLP), a data set can come in the form of sets of documents, while learning is performed at the sentence level. While assuming that the sentences are independent is inaccurate, it is reasonable to assume that the dependency between sentences decreases when

their distance in a document increases.

The remaining parts of the paper are structured as follows. We discuss some related work in Section 2 and present the problem formulation in Section 3. We present our main results on generalization bounds in Section 4, which are extended to learning with dependent examples in Section 5. We conclude the paper in Section 6.

## 2 Related Work

We first review some work on structured output prediction. Many algorithms have been developed to solve structured output prediction problems. Early techniques considered generative probabilistic models (e.g., hidden Markov models [Rabiner and Juang, 1986]). Motivated by the success of support vector machines (SVM), large-margin models for structured data were proposed in [Taskar *et al.*, 2003; Tsochantaridis *et al.*, 2005]. To reduce the model complexity, conditional random fields (CRFs) [Lafferty *et al.*, 2001] model the conditional distribution of the structured outputs rather than modeling the joint probability of the input and output. A key property of these models is that their prediction step can be viewed as maximising a scoring function. Such a scoring function enjoys a decomposition over the substructure so that the maximisation can be done efficiently. CRFs were combined with convolutional neural networks (CNNs) in [Chen *et al.*, 2017] to approach semantic segmentation problems, achieving better performance than CNNs alone.

In [Collins, 2001; Taskar *et al.*, 2003], the authors showed a generalization bound for their proposed models. However, they considered restricted models and losses (Hamming loss). A PAC-Bayesian bound is proved in [McAllester, 2007] for Bayesian prediction algorithms. In [Cortes *et al.*, 2016] the authors introduced a more general generalization bound that applies to general losses and models. Their bound scales as the square root of the number of classes. This can lead to vacuous bounds when the number of classes per substructure and their dependence on each other continue to increase.

[Ciliberto *et al.*, 2016] introduced the *implicit embedding* approach to structured output prediction where the label is encoded into a vector in some Hilbert space via an encoding function. A decoding function is also defined so that prediction is performed by composing a regression function and the decoding function, thus establishing a connection between structured output prediction and regression. They provided generalization bounds of the order of $O(m^{-\frac{1}{4}})$, where $m$ is the number of samples, which can be a problem for large $m$. Recently, [Ciliberto *et al.*, 2019] introduced the setting of localized structured output prediction, where they assume a form of weak dependence between substructures. Their model utilizes such assumption by treating each part of the structure as an independent sample. They prove bounds of the order $O((ml)^{-\frac{1}{4}})$ for their method, where $l$ is the number of substructures, under weakly dependent samples.

We now review the related work for multi-class classification (MCC), which is a specific case of structured output prediction. Various capacity measures of function classes were used to study generalization bounds of MCC, including Rademacher complexities [Lei *et al.*, 2015; Maurer, 2016; Li *et al.*, 2018; Maximov *et al.*, 2018; Musayeva *et al.*, 19], covering numbers [Zhang, 2004; Lei *et al.*, 2019; Ledent *et al.*, 2019] and the fat-shattering dimension [Guermeur, 2017]. While initial analyses implied generalization bounds with at least a linear dependency on the number of classes [Koltchinskii and Panchenko, 2002], the couplings among class components were exploited recently to get a dependency that can be as mild as square-root [Cortes *et al.*, 2016; Li *et al.*, 2018] or even logarithmic [Lei *et al.*, 2019; Wu *et al.*, 2021].

## 3 Problem Formulation

SOP refers to machine learning problems with an internal structure in the outputs (and potentially also in the inputs). For example in sequence-to-sequence prediction, both the input and output are sequences. In syntax analysis, the inputs are sequences of words and the output is a parse tree.

Let $\mathcal{X}$ be an input space (e.g., sentences in a given language) and $\mathcal{Y}$ be an output space (e.g., POS tags for the input sentences). In structured output prediction, the output space can often be decomposed into a number of substructures. Take POS tags as an example, where each word tag represents a substructure and the sequence of tags constitutes the structured output. Formally we define $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_l$, where $\mathcal{Y}_k$ is the set of possible classes a substructure $k$ can take. For a point $(x, y) \in \mathcal{X} \times \mathcal{Y}$, let $y^k$ denote the $k$-th element in $y$ (i.e., $y = (y^1, \ldots, y^l)$).

We aim to learn a scoring function $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ based on which we can perform the prediction as $\hat{y}(x) = \arg\max_{y \in \mathcal{Y}} h(x, y)$. The score function in structured output prediction can be described via a factor graph $G = (V, F, E)$, where $V = [l] := \{1, \ldots, l\}$ is the set of variable nodes, $F$ is a set of factor nodes, and $E$ is a set of undirected edges between a variable node and a factor node. Let $\mathcal{N}(f)$ be the set of nodes connected to the factor $f$ by an edge and $\mathcal{Y}_f = \Pi_{k \in \mathcal{N}(f)} \mathcal{Y}_k$. For brevity, we assume that $|\mathcal{Y}_f| = d$ for all $f$, where $|\mathcal{Y}_f|$ denotes the cardinality of $\mathcal{Y}_f$. Now we define the scoring function $h(x, y)$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ as

$$h(x, y) = \sum_{f \in F} h_f(x, y_f),$$

where $y_f := \{y^j : j \in \mathcal{N}(f)\}$ and $h_f : \mathcal{X} \times \mathcal{Y}_f \to \mathbb{R}$. Figure 1 gives an example of factor graphs and scoring functions.

Let $S = \{(x_i, y_i)\}_{i=1}^m$ be a training set with $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ being independently drawn from a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. We use a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ to measure the performance of prediction models, based on which we can define the margin loss [Cortes *et al.*, 2016] as $L_\rho : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \mathbb{R}$:

$$L_\rho(x, y, h) = \Phi^*(\max_{y' \neq y} \{L(y', y) - \frac{1}{\rho}[h(x, y) - h(x, y')]\}), \tag{1}$$

where $\Phi^*(r) = \min(M, \max(0, r))$, $M = \max_{y,y'} L(y, y')$ and $\mathcal{H} \subset \{h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}\}$ is some hypothesis class. Note that $L_\rho(x, y, h) \geq L(\hat{y}(x), y)$. Therefore, the obtained bounds for $L_\rho$ will also hold for $L$. We then define the population risk $R(h)$ and empirical risk $R_S(h)$ to quantify the

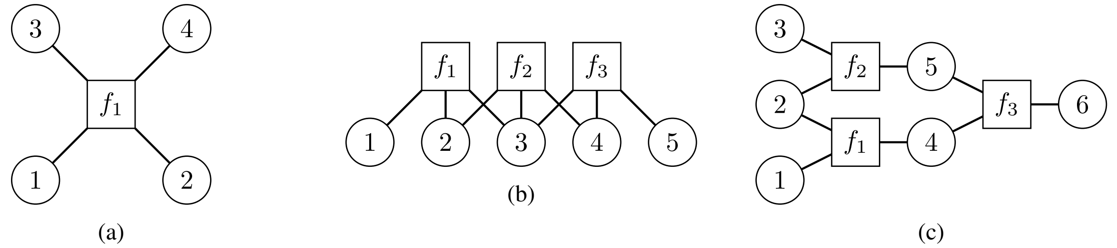

Figure 1: Examples of factor graphs. Panel (a) represents a factor graph with only one factor node. Note that $\mathcal{N}(f_1) = \{1, 2, 3, 4\}$ and $\mathcal{Y}_{f_1} = \mathcal{Y}_1 \times \mathcal{Y}_2 \times \mathcal{Y}_3 \times \mathcal{Y}_4$. If $\mathcal{Y}_i = \{1, 2, 3\}$ for all $i$, then $|\mathcal{Y}_{f_1}| = 3^4$. The corresponding scoring function is $h(x, y) = h_{f_1}(x, y^1, y^2, y^3, y^4)$. Panel (b) depicts an example of factor graph that assumes a sequence-like structure. The scoring function in this case is $h(x, y) = h_{f_1}(x, y^1, y^2, y^3) + h_{f_2}(x, y^2, y^3, y^4) + h_{f_3}(x, y^3, y^4, y^5)$. Panel (c) depicts an example of tree-like factor graph.

performance of a model $h$ on testing and training examples, respectively as:

$$R(h) = \mathbb{E}_{\mathcal{D}}[L_\rho(x, y, h)], \quad R_S(h) = \frac{1}{m}\sum_{i=1}^{m} L_\rho(x_i, y_i, h).$$

Let $\Psi$ be a feature function which maps an input-output example $(x, y) \in \mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}^D$, where $D$ is the dimension of feature space. In structured output prediction, the feature extractor takes a composite form according to the factor graph $G$, that is, $\Psi(x, y) = \sum_{f \in F} \Psi_f(x, y_f)$, where $\Phi_f : \mathcal{X} \times \mathcal{Y}_f \to \mathbb{R}$. We consider a linear scoring function $h^w(x, y) = \langle w, \Psi(x, y)\rangle$ indexed by a $w \in \mathbb{R}^D$. Then the hypothesis space becomes

$$\mathcal{H}_p = \big\{(x, y) \mapsto \langle w, \Psi(x, y)\rangle : \|w\|_p \le \Lambda, (x, y) \in \mathcal{X} \times \mathcal{Y}\big\}, \tag{2}$$

where $\|w\|_p = (\sum_{i=1}^{D} |w_i|^d)^{\frac{1}{p}}$ is the $\ell_p$-norm of $w = (w_1, \ldots, w_D)$. We also define the class of loss functions

$$F_{p,\Lambda,\rho} := \big\{(x, y) \mapsto L_\rho(x, y, h^w) : h^w \in \mathcal{H}_p\big\}. \tag{3}$$

# 4 Main Results

In this section, we present our main results on generalization bounds for structured output prediction. We consider two types of generalization bounds: complexity-based bounds and stability-based bounds. Our aim is to develop bounds with a very mild dependency on the size of the label set, thus laying a solid foundation for structured output prediction, where the size of label set $\mathcal{Y}$ is often extremely large in practice. A key discovery to both our stability-based and complexity-based analysis is to note the Lipschitz continuity of loss functions w.r.t. infinity-norm $\|\cdot\|_\infty$.

**Definition 1** (Lipschitz continuity)**.** We say that a loss function $L(x, y, h)$ is $(\tau, \ell_\infty)$-Lipschitz in the last argument if, for any $h, \tilde{h} \in \mathcal{H}$ and all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we have:

$$|L(x, y, h) - L(x, y, \tilde{h})| \le \tau \max_{y' \in \mathcal{Y}} |h(x, y') - \tilde{h}(x, y')|.$$

The existing analysis [Cortes *et al.*, 2016] uses the $(\tau_2, \ell_2)$ Lipschitz continuity of loss functions:

$$|L(x, y, h) - L(x, y, \tilde{h})| \le \tau_2 \Big(\sum_{y' \in \mathcal{Y}} |h(x, y') - \tilde{h}(x, y')|^2\Big)^{1/2}.$$

Note that the Lipschitz continuity w.r.t. $\ell_\infty$-norm is much stronger than that w.r.t. $\ell_2$-norm. Indeed, if $L$ is $(\tau, \ell_\infty)$-Lipschitz then it is also $(\tau, \ell_2)$ Lipschitz since $\|\cdot\|_\infty \le \|\cdot\|_2$. As a comparison, a $(\tau_2, \ell_2)$-Lipschitz function can be $(\tau_2\sqrt{|\mathcal{Y}|}, \ell_\infty)$-Lipschitz due to the norm relationship $\|\cdot\|_2 \le \sqrt{|\mathcal{Y}|}\|\cdot\|_\infty$ (the equality can hold in some cases).

In Lemma 1 we build the $\ell_\infty$-Lipschitz continuity of $L_\rho$ for structured output prediction. A remarkable property is that the involved Lipschitz constant is independent of $|\mathcal{Y}|$. This shows that the loss function in structured output prediction is well behaved in the sense of Lipschitz continuity. However, the existing analysis based on the $(\tau_2, \ell_2)$-Lipschitz continuity fails to exploit this strong regularity, and therefore only implies suboptimal bounds with at least a square-root dependency on the size of the label set. The proof of Lemma 1 below is given in the arXiv version.

**Lemma 1.** *The loss function $L_\rho$ is $(\frac{2}{\rho}, \ell_\infty)$-Lipschitz with respect to the scoring function $h$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.*

## 4.1 Complexity-based Generalization Bounds

We develop generalization bounds with high probability here. Our basic tool to this aim is the Rademacher complexity.

**Definition 2.** The empirical Rademacher complexity of a function class $\mathcal{H}$ of real-valued functions is defined as:

$$\mathfrak{R}_S(\mathcal{H}) = \mathbb{E}_\sigma[\sup_{f \in \mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(x_i)], \tag{4}$$

where $\{\sigma_i\}$ are random variables with equal probability of being either $+1$ or $-1$.

**Theorem 1.** *Let $\rho > 0$ be. Then the Rademacher complexity of the loss class $F_{p,\Lambda,\rho}$ is bounded as follows:*

$$\mathfrak{R}(F_{p,\Lambda,\rho}) \le \frac{4}{m} + \frac{144\sqrt{q-1}\Psi^*\Lambda|F|}{\rho\sqrt{m}}\tilde{L}, \tag{5}$$

*where* $\tilde{L} = \sqrt{\log(2md|F|[8\Psi^*\Lambda m|F|/\rho + 3] + 1)}\log(m)$, $\Psi^* = \sup_{f \in F, y \in \mathcal{Y}_f, x \in \mathcal{X}} \|\Psi_f(x, y)\|_q$, *and* $q = p/(p-1)$.

The proof strategy is to to relate the complexity of the loss class $F_{p,\Lambda,\rho}$ to a complexity of a scalar linear function class on an extended set of size $m|F|d$, thus moving contribution of $d$ to the complexity from the output dimension to the size of training set. We then utilize standard bounds [Zhang, 2002] that admit log dependency on the size of training set. The detailed proof is given in the arXiv version of the paper.

**Remark 1.** We now compare our results with related work. In [Cortes *et al.*, 2016], the authors bounded the Rademacher complexity of $F_{p,\Lambda,\rho}$ by a factor graph Rademacher complexity. Specifically for the loss class (3) they proved $\mathfrak{R}(F_{p,\Lambda,\rho}) \leq \frac{2\sqrt{2}}{\rho}\hat{\mathfrak{R}}^G_S(\mathcal{H}_p)$, where $\hat{\mathfrak{R}}^G_S(\mathcal{H}_p)$ is defined as

$$\frac{1}{m}\mathbb{E}_\epsilon\Big[\sup_{h\in\mathcal{H}} \sum_{i\in[m],f\in F,y\in\mathcal{Y}_f} \sqrt{|F|}\epsilon_{i,f,y}\,\langle w,\Psi_f(x_i,y)\rangle\Big].$$

Here $\epsilon = (\epsilon_{i,f,y})_{i\in[m],f\in F,y\in\mathcal{Y}_f}$ and each $\epsilon_{i,f,y}$ is an independent Rademacher variable. Combining the result from Theorem 2 in [Cortes *et al.*, 2016], we get the following bound for learning with $\ell_2$-regularization: $\mathfrak{R}(F_{2,\Lambda,\rho}) \leq O\Big(\frac{\Lambda\Psi^*|F|\sqrt{d}}{\rho\sqrt{m}}\Big)$. Note the bound exhibit a square-root dependence on the number of classes per factor $d = |\mathcal{Y}_f|$. Thus it is vacuous for typical SOPPs, where the number of class labels grows exponentially w.r.t. the size of the output. For comparison, our bounds enjoy a log dependency on $d$ and therefore still imply meaningful generalization bounds in this setting.

As a direct corollary, we use the connection between generalization and Rademacher complexity to get Theorem 2.

**Theorem 2** (Generalization Bounds)**.** *For any $\rho > 0$, $\delta \in (0,1)$, and $h \in \mathcal{H}_p$, with probability at least $1-\delta$ over the draw of training data $S$, the following bound holds:*

$$R(h) \leq R_S(h) + \frac{8}{m} + \frac{288\sqrt{q-1}\Psi^*\Lambda|F|}{\rho\sqrt{m}}\tilde{L} + M\sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

## 4.2 Stability-based Generalization Bounds

In this section, we present generalization bounds in expectation for structured output prediction by leveraging the lens of algorithmic stability. Algorithmic stability is a fundamental concept in statistical learning theory, which measures the sensitiveness of output models when the training dataset of an algorithm $A$ is slightly perturbed. For any algorithm $A$, we use $A(S)$ to denote the model produced by running $A$ over the training examples $S$.

**Definition 3** (Uniform Stability)**.** A stochastic algorithm $A$ is $\epsilon$-uniformly stable if, for all training datasets $S, \widetilde{S} \in \mathcal{Z}^n$ that differ by at most one example, we have

$$\sup_{x,y}\mathbb{E}_A\big[L_\rho(x,y,A(S)) - L_\rho(x,y,A(\widetilde{S}))\big] \leq \epsilon. \qquad (6)$$

Algorithmic stability naturally implies quantitative generalization bounds, as shown in the following lemma [Shalev-Shwartz *et al.*, 2010].

**Lemma 2** (Generalization via uniform stability)**.** *Let $A$ be $\epsilon$-uniformly stable. Then $\big|\mathbb{E}_{S,A}\big[R(A(S)) - R_S(A(S))\big]\big| \leq \epsilon$.*

We will apply algorithmic stability to study two representative algorithms for SOP: regularization and stochastic gradient descent. For brevity, we use the abbreviation $R(w) = R(h^w), R_S(w) = R_S(h^w)$, etc. We also write $w^* = \arg\inf_w R(w)$ for the minimizer of the population risk.

### Regularized Risk Minimization

RRM is a popular scheme to overcome overfitting in machine learning. The basic idea is to add a regularizer to the empirical risk and build a regularized empirical risk $R_S^\lambda$. Then we minimize the resulting objective function to obtain a model $w_S$ as follows:

$$w_S = \arg\min_w \big[R_S^\lambda(h^w) := R_S(h^w) + \frac{\lambda}{2}\|w\|_2^2\big]. \qquad (7)$$

Here we omit the dependency of $w_S$ on the regularization parameter for brevity. In the following lemma to be proved in the arXiv version, we show that the above regularization algorithm is uniformly stable. Let $\kappa := \sup_{x,y}\|\Psi(x,y)\|_2$.

**Lemma 3.** *Let $A$ be defined as* (7)*, i.e., $A(S) = w_S$. Then $A$ is $\frac{16\kappa^2}{m\rho^2\lambda}$-uniformly stable.*

This lemma is a variant of the stability bound in [Bousquet and Elisseeff, 2002], which, however, requires the loss function to be admissible. We adapt their technique to the setting of structured output prediction and a key step in our analysis is again the Lipschitz continuity of the loss function w.r.t. the $\ell_\infty$ norm. A use of the classical Lipschitz continuity w.r.t. $\ell_2$ norm would incur a bound with at least a square-root dependency on $d$. For comparison, the consideration of Lipschitz continuity w.r.t. the $\ell_\infty$ norm allows us to get stability bounds independent of the size of the label set.

We can combine the Lipschitz continuity of loss functions, the stability of regularization schemes established in Lemma 3 and Lemma 2 together to get the following generalization bounds for structured output prediction. Let

$$w^\lambda = \arg\inf_w \big[R^\lambda(w) := R(h^w) + \frac{\lambda}{2}\|w\|_2^2\big]$$

be the minimizer of the regularized risk. We have the following result, whose proof is given in the arXiv version.

**Theorem 3.** *Let $w_S$ be defined in* (7)*. Then*

$$\mathbb{E}\big[R^\lambda(w_S) - R^\lambda(w^\lambda)\big] \leq \frac{16\kappa^2}{m\rho^2\lambda}. \qquad (8)$$

*Furthermore, if we choose $\lambda = \frac{4\sqrt{2}\kappa}{\sqrt{m}\rho\|w^*\|_2}$, then*

$$\mathbb{E}\big[R(w_S)\big] - R(w^*) \leq \frac{4\sqrt{2}\kappa\|w^*\|_2}{\sqrt{m}\rho}. \qquad (9)$$

### Stochastic Gradient Descent

We now turn to the performance of SGD for structured output prediction. SGD is a popular optimization algorithm with wide applications in learning in a big data setting. Let $w^{(1)}$ be the initial point and $\{\eta_t\}$ be a sequence of positive step sizes. At the $t$-th iteration, we first randomly select an index $i_t$ according to the uniform distribution over $[m]$, which is used to build a stochastic gradient $L'_\rho(x_{i_t}, y_{i_t}, h^{w^{(t)}})$ ($L'_\rho(x_{i_t}, y_{i_t}, h^{w^{(t)}})$ denotes a subgradient of $L_\rho(x_{i_t}, y_{i_t}, h^w)$ at $w = w^{(t)}$). Then we update the model along the negative direction of the stochastic gradient

$$w^{(t+1)} = w^{(t)} - \eta_t L'_\rho(x_{i_t}, y_{i_t}, h^{w^{(t)}}). \qquad (10)$$

This scheme of selecting a single example to build a stochastic gradient allows SGD to get sample-size independent iteration complexity, and is especially appealing if $m$ is large. Since we consider a linear scoring function $h^w$, the loss function $L_\rho$ is convex w.r.t. $w$. In the following lemma to be proved in the arXiv version, we build the uniform stability of SGD for structured output prediction. Note here we do not require the loss function to be smooth [Lei and Ying, 2020].

**Lemma 4.** *Let $S = \{z_1, \ldots, z_m\}$ and $S' = \{z'_1, \ldots, z'_m\}$ be two datasets that differ only by a single example. Let $\{w^{(t)}\}$ and $\hat{w}^{(t)}$ be two sequences produced by SGD based on $S$ and $S'$, respectively. Then*

$$\mathbb{E}_A\big[\|w^{(t+1)} - \hat{w}^{(t+1)}\|_2^2\big] \le 16e(1 + t/m^2)\kappa^2\rho^{-2}\sum_{j=1}^{t}\eta_j^2.$$

According to Lemma 4, we know that the algorithm becomes more and more unstable as we run more and more iterations. We can use this stability bound to derive generalization bounds of SGD for structured output prediction. The proof is given in the arXiv version.

**Theorem 4.** *Let$\{w^{(t)}\}$ be produced by* (10) *with $\eta_t = \eta$. Then*

$$\mathbb{E}\big[R(\bar{w}^{(T)})\big] - R(w^*) \le O\Big((\sqrt{T} + T/m)\kappa^2\eta + \frac{1 + T\kappa^2\eta^2}{T\eta}\Big), \tag{11}$$

*where $\bar{w}^{(T)} = \frac{1}{T}\sum_{t=1}^{T} w^{(t)}$ is an average of iterates.*

The upper bound (11) involves two terms. The first term $\sqrt{T} + T/m$ comes from controlling the generalization error $R(\bar{w}^{(T)}) - R_S(\bar{w}^{(T)})$, while the second term $\frac{1+T\eta^2}{T\eta}$ comes from controlling the optimization error $R_S(\bar{w}^{(T)}) - R_S(w^*)$. It is clear the optimization error decreases w.r.t. $T$, while the generalization error grows in the learning process. Therefore, we need to trade-off these two terms by early-stoping SGD as done by the following corollary. We write $B \asymp \widetilde{B}$ if there are absolute constants $c_1$ and $c_2$ such that $c_1 B \le \widetilde{B} \le c_2 B$.

**Corollary 1.** *Let $\{w^{(t)}\}$ be the sequence produced by* (10) *with $\eta_t = \eta$. If we choose $T \asymp m^2$ and $\eta \asymp T^{-\frac{3}{4}}/\kappa$, then*

$$\mathbb{E}\big[R(\bar{w}^{(T)})\big] - R(w^*) \le O(\kappa m^{-\frac{1}{2}}).$$

**Remark 2.** According to Theorem 3 and Corollary 1, we know that both the regularization method and SGD are able to achieve the generalization bound $O(1/\sqrt{m})$, which is minimax optimal. While RRM requires the objective function to be strongly convex, SGD only requires the objective function to be convex. Remarkably, these generalization bounds do not admit any dependency on the size of the label set, and provide a convincing explanation on why SOP often works well even if the problem has more class labels than training examples. To our best knowledge, these are the first label-size free generalization bounds. As compared to Theorem 2 on high-probability bounds, our generalization bounds here are stated in expectation. It should be noted that our bounds in expectation require the loss functions to be convex, while the high-probability analysis also applies to nonconvex cases.

### 4.3 Applications

In this section we discuss applications of our bounds and compare them to those of [Cortes *et al.*, 2016].

**Example 1.** Consider pair-wise Markov networks with fixed number of substructures $l$ [Taskar *et al.*, 2003]. Specifically, we have $\mathcal{Y} = \mathcal{Y}_1 \times \ldots \times \mathcal{Y}_l$ and $\mathcal{Y}_k \in [c]$ for $k \in [l]$. Further, we have sequence-like connections, i.e., there is an arrangement of output nodes such that if a factor $f \in F$ is connected to two nodes then they are neighbors in that arrangement. Therefore we have $|F| = l - 1$ and $d = c^2$. We further assume an unnormalized hamming loss $L(y, y') = \sum_{k=1}^{l} \mathbb{I}_{y_k \neq y'_k}$ so that we normalize later in the bound to get rid of the dependence on $l = |F| + 1$. For regularized learning with these Markov networks, the Rademacher complexity of loss function classes was bounded in [Cortes *et al.*, 2016] as $\mathfrak{R}(F_{2,\Lambda,\rho}) \le O(\frac{\Lambda\Psi^* c}{\rho\sqrt{m}})$. As a comparison, our Rademacher complexity bound in Theorem 1 reduces to an upper bound on $\mathfrak{R}(F_{2,\Lambda,\rho})$ that has the form

$$O\left(\frac{\Lambda\Psi^* \log m\sqrt{\log(2mc^2 l[8\Psi^*\Lambda m/\rho + 3] + 1)}}{\rho\sqrt{m}}\right).$$

Therefore, our bound significantly outperforms their bound by dropping their linear dependency on $c$ to a logarithmic dependency. If we further extend the model so that each factor $f$ is connected to $v$ nodes instead of 2, their bound grows, as a function of $v$, as $O(c^{v/2})$ while ours increase only $O(\sqrt{v})$. Furthermore, according to Theorems 3, 4, we can get generalization bounds $O(\kappa/\sqrt{m})$ in expectation for both RRM and SGD, where the log dependency is further removed.

**Example 2.** As the second example we consider multi-class classification. In this case we have no substructures and therefore $|F| = 1, \mathcal{Y}_1 = \mathcal{Y}$ where $\mathcal{Y} = [c]$, $d = c$. In [Cortes *et al.*, 2016], the Rademacher complexity for multi-class learning with $\ell_2$ regularization was shown to satisfy $\mathfrak{R}(F_{2,\Lambda,\rho}) \le O\left(\frac{\Psi^*\Lambda\sqrt{c}}{\rho\sqrt{m}}\right)$.

Our analysis instead shows $\mathfrak{R}(F_{2,\Lambda,\rho})$ is bounded by

$$O\left(\frac{\Psi^*\Lambda\sqrt{\log(2mc[8\Psi^*\Lambda m/\rho + 3] + 1)}\log m}{\rho\sqrt{m}}\right).$$

It is clear that we drop the square root dependency in $c$ in [Cortes *et al.*, 2016] to a log dependence. Analogous to Example 1, the log dependency can be further removed if we consider generalization bounds in expectation, as shown in Theorems 3 and 4.

**Example 3.** In this example we explore the possibility of combining SOP models above with a learned feature extraction function $\Psi$ as was practically explored in [Chen *et al.*, 2017; Hinton *et al.*, 2012]. Consider the case where $\Psi$ is a CNN that takes $x$ as input and outputs different $D$-dimensional vector $\Psi_f(x, y_f)$ for each factor $f$ and label $y_f$. Chaining the covers, one can bound the Rademacher complexity of the combined class as $\widetilde{O}\left(\frac{\sqrt{q-1}\Psi^*\Lambda|F|}{\rho\sqrt{m}}\right) + \widetilde{O}\left(\sqrt{\frac{\bar{D}}{m}}\log(\tilde{G})\right)$, where the notation $\widetilde{O}$ hides logarithmic factors, $\bar{D}$ is the number of network parameters and $\tilde{G}$ is a product of norms of network weight matrices. The derivation of this bound is given in the arXiv version.

## 5 Learning Weakly Dependent Sequences

In the above bounds we assumed that the examples are sampled independently from each other. However, this assumption is often violated in practice. For example, consider the problem of POS tagging. We are usually given a dataset of documents each of which contains a sequence of sentences. There are two natural assumptions. (1) We may assume that each document is a long sequence of dependent words. This assumption is too pessimistic. The considered sample size becomes too small, and the prediction complexity increases while, as sentences get further apart, the dependence between them decreases, and thus the effective sample size increases. (2) We may assume that each sentence is independent of the others within and across documents. This assumption on the other hand is too optimistic. Sentences following each other in the same document indeed have some degree of dependence. We formalize this dependence in a hierarchical manner, thus providing a trade-off between these two assumptions. Namely, we assume that the documents are independent of each other while sentences within a document are only weakly dependent. We note that the term document here does not necessarily mean an actual text document but rather any sequence of examples (e.g., for a dataset of videos, one video is a document as it is a sequence of images).

We now formalize the idea above. We are given a training set of independent documents $\{D_i\}_{i=1}^m$. Each document $D_i$ is a sequence of weakly dependent examples $D_i = (z_i^j)_{j=1}^J$. Since the structured output prediction framework in the above section subsumed the usual classification paradigm, we assume that the sequence elements classes follows it. That is, $z_j^i \in \mathcal{X} \times \mathcal{Y} =: \mathcal{Z}$, where $\mathcal{X}$ and $\mathcal{Y}$ defined as above.

Now we define precisely how the examples within each document $D_i$ are weakly dependent. We assume that each example within a given document is sampled from a $\beta$-mixing process, defined below, at times $1, 2, \ldots, J$.

**Definition 4** (Stationary $\beta$-mixing Stochastic Process)**.** Let $(z^k)_{k=-\infty}^{\infty}$ be a stationary stochastic process and $\sigma_L = \sigma((z^k)_{k=1}^L)$ and $\sigma_{L+a} = \sigma((z^k)_{k=L+a}^{\infty})$ be the sigma algebras generated by the random variables $Z_1^L = (z^1, \ldots, z^L)$ and $z_{L+a}^{\infty} = (z^{L+a}, \ldots, Z^{\infty})$. Define the $\beta$-mixing coefficient $\beta(a) := \sup_{L \geq 1} \mathbb{E}\left[\sup_{B \in \sigma_{L+a}} |P(B|\sigma_L) - P(B)|\right]$. The process is called $\beta$-mixing if $\lim_{a \to \infty} \beta(a) = 0$. It is called exponentially mixing if $\beta(a) \leq \beta_0 \exp(-\beta_1 a^r)$ or algebraically mixing if $\beta(a) \leq \beta_0 / a^r$, for positive $\beta_0$, $\beta_1$ and $r$.

Some examples of exponentially mixing process include a class of Autoregressive Moving Average (ARMA) [Mokkadem, 1988] and a class of Markov process [Rosenblatt, 2012].

Now let $\mathcal{H}$ be a structured output prediction function class as defined in the previous section (see (2)). For $h \in \mathcal{H}$, let $L_h : \mathcal{Z} \to [0, M]$ be a loss function over elements of the sequence $z^k$. An example for such a loss is given in equation (1). Again we are interested in high probability bounds on the difference between two quantities: the empirical risk $\hat{R}_S(h)$ and the true risk $R(h)$, which are defined as $R_S(h) = \frac{1}{mJ} \sum_{i=1}^m \sum_{j=1}^J L_h(z_i^j)$ and $R(h) = \mathbb{E}_S[R_S(h)]$.

Theorem 5 summarizes the main results of this section.

**Theorem 5.** *Let $F_{p,\Lambda,\rho}$ be the loss class defined in* (3) *and let $S$ be a set of independent and identically distributed documents $D_i$, $i = 1, \ldots, m$, where each document is a sequence of examples $(z_i^j)$ $j = 1, \ldots, J$ drawn from a $\beta$-mixing process. For any integer $a > 0$ such that $J$ is a multiple of $2a$. Let $\delta > 2m(\frac{J}{2a} - 1)\beta(a)$, then with probability at least $1 - \delta$, the following inequality holds uniformly over all $h \in \mathcal{H}_p$*

$$\begin{aligned} R(h) \leq& \hat{R}_S(h) + O\left(\frac{\sqrt{2a(q-1)}\Psi^*\Lambda|F|}{\rho\sqrt{mJ}}\tilde{L}\right) \\ &+ \frac{M\sqrt{a}\sqrt{\log\left(\frac{2}{\delta - 2m(\frac{J}{2a}-1)\beta(a)}\right)}}{\sqrt{mJ}}, \end{aligned} \tag{12}$$

*where $\tilde{L} = \sqrt{\log(2mJd|F|[8\Psi^*\Lambda mJ/\rho + 3] + 1)}\log(mJ)$.*

**Remark 3.** Note that the bound unsurprisingly depends on the same main quantities as the bound in Theorem 2. To better interpret it consider the following two extreme cases. (1) The elements inside each document are independent of each other. Note that in this case $\beta(a) = 0$, for all $a$, hence $a$ can be chosen to be 1 and the bound boils down to the bound in Theorem 2. (2) The elements inside each document are strongly dependent. Thus, $\beta(a) \approx 1$ for all $a$ and therefore selecting $a = \frac{J}{2}$ leads to the bound in Theorem 2 with only $m$ training examples. We further note that $\beta(a) \to 0$ as $a \to \infty$, therefore, for any process admitting a fast decaying $\beta(a)$ the term $2m(\frac{J}{2a} - 1)\beta(a)$ approaches 0 fast for moderate $a$.

## 6 Conclusion

In this paper, we advance the state of the art in the generalization analysis of structured output prediction. We consider two types of generalization bounds: complexity-based and stability-based bounds. Our complexity-based approach produces bounds with high probability that admit a log dependency on the size of the label set. The stability-based approach further removes this log dependency for generalization bounds in expectation. This significantly improves the existing bounds, which have at least a square root dependency. We also extend our discussion to the setting of learning with weakly dependent training examples.

A very interesting question is to investigate whether the log dependency in the high probability analysis is an artefact of our analysis or is really essential. Another question is to extend our generalization bounds in expectation to learning with nonconvex functions.

## Acknowledgments

MK, AL and WM acknowledge support by the German Research Foundation (DFG) award KL 2698/2-1 and by the German Federal Ministry of Science and Education (BMBF) awards 01IS18051A, 031B0770E, and 01MK20014U. YL acknowledges support by NSFC under Grant No. 61806091.

## References

[Bousquet and Elisseeff, 2002] Olivier Bousquet and André Elisseeff. Stability and generalization. *JMLR*, 2:499–526, 2002.

[Chen *et al.*, 2017] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *TPAMI*, 40(4):834–848, 2017.

[Ciliberto *et al.*, 2016] Carlo Ciliberto, Alessandro Rudi, and Lorenzo Rosasco. A consistent regularization approach for structured prediction. In *NeurIPS*, 2016.

[Ciliberto *et al.*, 2019] Carlo Ciliberto, Francis Bach, and Alessandro Rudi. Localized structured prediction. In *NeurIPS*, 2019.

[Collins, 2001] Michael Collins. Parameter estimation for statistical parsing models: Theory and practice of. In *IWPT*, pages 4–15, 2001.

[Cortes *et al.*, 2016] Corinna Cortes, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. Structured prediction theory based on factor graph complexity. In *NeurIPS*, 2016.

[Guermeur, 2017] Yann Guermeur. Lp-norm sauer–shelah lemma for margin multi-category classifiers. *Journal of Computer and System Sciences*, 89:450–473, 2017.

[Hinton *et al.*, 2012] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[Koltchinskii and Panchenko, 2002] Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, pages 1–50, 2002.

[Lafferty *et al.*, 2001] John D Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[Ledent *et al.*, 2019] Antoine Ledent, Waleed Mustafa, Yunwen Lei, and Marius Kloft. Norm-based generalisation bounds for multi-class convolutional neural networks. *CoRR*, abs/1905.12430, 2019.

[Lei and Ying, 2020] Yunwen Lei and Yiming Ying. Fine-grained analysis of stability and generalization for stochastic gradient descent. In *ICML*, 2020.

[Lei *et al.*, 2015] Yunwen Lei, Urun Dogan, Alexander Binder, and Marius Kloft. Multi-class svms: From tighter data-dependent generalization bounds to novel algorithms. In *NeurIPS*, 2015.

[Lei *et al.*, 2019] Yunwen Lei, Ürün Dogan, Ding-Xuan Zhou, and Marius Kloft. Data-dependent generalization bounds for multi-class classification. *IEEE Transactions on Information Theory*, 65(5):2995–3021, 2019.

[Li *et al.*, 2018] Jian Li, Yong Liu, Rong Yin, Hua Zhang, Lizhong Ding, and Weiping Wang. Multi-class learning: from theory to algorithm. In *NeurIPS*, 2018.

[Lucchi *et al.*, 2013] Aurelien Lucchi, Y. Li, and P. Fua. Learning for structured prediction using approximate subgradient descent with working sets. In *CVPR*, 2013.

[Maurer, 2016] Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *ALT*, 2016.

[Maximov *et al.*, 2018] Yury Maximov, Massih-Reza Amini, and Zaid Harchaoui. Rademacher complexity bounds for a penalized multi-class semi-supervised algorithm. *JAIR*, 61:761–786, 2018.

[McAllester, 2007] David McAllester. Generalization bounds and consistency. *Predicting structured data*, pages 247–261, 2007.

[Mokkadem, 1988] Abdelkader Mokkadem. Mixing properties of arma processes. *Stochastic Processes and Their Applications*, 29(2):309–315, 1988.

[Musayeva *et al.*, 19] Khadija Musayeva, Fabien Lauer, and Yann Guermeur. Rademacher complexity and generalization performance of multi-category margin classifiers. *Neurocomputing*, 342:6–15, 19.

[Rabiner and Juang, 1986] Lawrence Rabiner and B Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[Rosenblatt, 2012] Murray Rosenblatt. *Markov Processes, Structure and Asymptotic Behavior: Structure and Asymptotic Behavior*, volume 184. Springer Science & Business Media, 2012.

[Shalev-Shwartz *et al.*, 2010] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *JMLR*, 11:2635–2670, 2010.

[Taskar *et al.*, 2003] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. *NeurIPS*, 16, 2003.

[Tsochantaridis *et al.*, 2005] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005.

[Vinyals *et al.*, 2015] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[Wu *et al.*, 2021] Liang Wu, Antoine Ledent, Yunwen Lei, and Marius Kloft. Fine-grained generalization analysis of vector-valued learning. In *AAAI*, 2021.

[Zhang, 2002] Tong Zhang. Covering number bounds of certain regularized linear function classes. *JMLR*, 2, 2002.

[Zhang, 2004] Tong Zhang. Statistical analysis of some multi-category large margin classification methods. *JMLR*, 5:1225–1251, 2004.