Abstract

Structured prediction tasks contain complex dependencies between multi-dimensional labels that can be utilized for improving accuracy and creating consistent predictions. Recently, neural networks have been used for learning dependencies in an unconstrained manner by defining an energy function over the input and labels, relying on gradient-based optimization for inference. Instead, we propose usage of an auxiliary adversarial loss to enforce structure on the outputs of deep models. In addition to a local per label loss, the model is trained to fool a discriminator, which in turn is optimized to distinguish between true labels and the model’s predictions. This adaptive loss regularizes the model towards the prior joint label distribution, enabling learning of global structure and implicit constraints. We focus on the multi-label classification task, and show that our method achieves state-of-the-art results on common benchmarks, with significantly faster inference compared to current approaches.

1. Introduction

Numerous tasks of significant interest require learning a mapping from inputs to structured multivariate outputs (e.g., semantic image segmentation, sequence tagging, and multi-label classification). Structured prediction is a core problem in machine learning, that aims to improve prediction quality by utilizing dependencies between the output variables in such tasks. Modeling high-dimensional distributions is a challenging task, with various existing approaches that trade-off between capacity and computational tractability.

As part of the surge in deep learning research, techniques for incorporating deep architectures into existing structured prediction paradigms were proposed. One popular approach is replacing linear functions with neural networks for modeling factorized scores in graphical models, harnessing their representational power (Chen et al., 2014; 2015; Schwing & Urtasun, 2015; Zheng et al., 2015; Ma & Hovy, 2016). Despite proving to be effective on multiple tasks, the necessity of simple graph structures for tractable inference restricts these methods from fully capturing intricate interactions.

A different approach, initiated by the introduction of Structured Prediction Energy Networks (SPENs) (Belanger & McCallum, 2016), aims to resolve said structural limitations through usage of neural networks to define a global energy function over the inputs and structured outputs. Deep architectures are capable of implicitly capturing dependencies and constraints on labels that would otherwise form intractable graphical models. The increase in capacity comes at a cost of an expensive inference phase, which relies on usage of gradient descent with respect to the labels. Further work has mainly focused on different training objectives, yielding promising results on several tasks, such as semantic role labeling and image segmentation (Belanger et al., 2017; Gygli et al., 2017).

In this work, our goal is to combine deep learning and structured prediction while maintaining a balance between inference cost and model capacity. To this end, we propose usage of a global loss function, which takes into account all predicted labels, for learning structure and constraints implicitly. A deep neural network classifier is trained with two minimization objectives: a local per label discriminative loss and a global structural loss. Choosing a fitting loss for the latter is a crucial part of our approach, as it is responsible for enforcing the classifier to create predictions that obey existing dependencies between the labels. Manually defining a structural loss is a demanding task since it typically requires specific domain expertise per application. Preferably, we aim to avoid this necessity through an adaptive and generic loss.

Generative adversarial networks (Goodfellow et al., 2014) have been shown effective in implicitly modeling complex multivariate distributions. Training opposing generator and discriminator networks in an alternating fashion led to impressive results in realistic image generation (Brock et al., 2018). Recent successes of GANs motivate adopting an adversarial loss as the global structure loss in our method - Adversarial Loss as Prior Regularization (ALPR). The classifier is analogous to the generator, generating multi-
Adversarial Loss as Prior Regularization for Structured Prediction

In the following section we define the relevant setup, introducing a few notations and the main components of our method.

We consider the task of assigning for each \( x \in \mathcal{X} \) an output \( y \in \mathcal{Y} \), where \( y \) can have a complex multivariate structure. We focus on the binary multi-label classification task where \( \mathcal{Y} = \{0, 1\}^m \). As in Belanger & McCallum 2016, we relax the discreteness constraint on the labels by replacing the original label space with \([0, 1]^m\), allowing differentiation of the adversarial loss through the continuous outputs of the classifier.

Our structured prediction adversarial framework, ALPR, consists of two competing deep neural network models, as shown in Figure 1: A classifier \( C_\theta : \mathcal{X} \rightarrow \mathcal{Y} \) that generates candidate predictions \( \hat{y} \) for a given input, and a discriminator \( D_\phi : \mathcal{Y} \rightarrow [0, 1] \) that estimates the probability that a label assignment was sampled from the true joint label distribution \( p \). We denote the classifier and discriminator parameters by \( \theta \) and \( \phi \), respectively, and the distribution of \( C_\theta(x) \) with \( p_\theta \). Thus, for a given \( y \) the output of the discriminator, \( D_\phi(y) \), measures its confidence that \( y \) was sampled from \( p \) as opposed to \( p_\theta \). We note that sampling from \( p_\theta \) is equivalent to classifying with \( C_\theta \) a randomly sampled \( x \sim p \).

Finally, let \( \ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+ \) be a local discriminative loss, limited to summation interactions between independent per label losses \( \ell_i \). In our case we take each \( \ell_i \) to be the binary cross-entropy loss:

\[
\ell_{\text{bce}}(\hat{y}_i, y_i) = -y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i)
\]

3. Adversarial Loss for Structured Prediction

We now describe in detail our adversarial approach for structured prediction. ALPR is conceptually simple, augmenting solely the training procedure of a deep neural network classifier \( C_\theta \) by adding an auxiliary adversarial loss. Inference for new inputs requires only a single application of \( C_\theta \) to generate continuous label predictions. Then, producing the final binary outputs is done with a predetermined threshold, tuned on a development set. The discriminator \( D_\phi \) is only dimensional predictions instead of images, and the discriminator guides it towards matching the prior joint label distribution in a generic and extensible manner. The adversarial loss endows the model with data-driven structure learning capabilities, which allow avoiding assignments that might seem likely for each independent label, but jointly form an impossible assignment that a competent discriminator would detect.

After training, inference is reduced to a single forward pass of the classifier, replacing the expensive iterative inference of energy-based architectures. Meanwhile, we also sidestep the need for manually defining possible label interactions, relying on a competent adversary for enabling structure learning. We demonstrate the effectiveness of our method on standard binary multi-label classification benchmarks. Empirical results show that our adversarial setup surpasses currently existing approaches, substantially improving the performance of a standard MLP classifier. Our code implementation based on the PyTorch framework is attached.
Adversarial Loss as Prior Regularization for Structured Prediction

part of the optimization process and is not necessary during evaluation.

Though we limit the following description and evaluation to the binary case, ALPR is easily applicable to non-binary labels and other domains. Mainly, it is only necessary to adjust the architectures of \( C_\theta \) and \( D_\phi \) according to the task at hand.

3.1. Optimization

Training of both models in the ALPR framework is done simultaneously, similarly to the standard GAN alternating optimization presented in Goodfellow et al. 2014. We train the discriminator to distinguish between predicted and true labels through maximizing the following objective with respect to \( \phi \):

\[
V(\theta, \phi) = \mathbb{E}_{y \sim p} [\log D_\phi(y)] + \mathbb{E}_{x \sim p} [\log(1 - D_\phi(C_\theta(x))] \quad (2)
\]

In contrast, the classifier is trained to minimize the objective above with respect to \( \theta \). We note that since the first term in \( V \) does not depend on \( \theta \), it can be safely ignored. The final classifier multitask loss is obtained by combining the auxiliary adversarial loss and the local discriminative loss via a scaling hyperparameter \( \lambda_{adv} \):

\[
\mathcal{L}(\theta) = \mathcal{L}_{bce}(\theta) + \lambda_{adv} \mathcal{L}_{adv}(\theta) \quad (3)
\]

with \( \mathcal{L}_{bce}(\theta) = \mathbb{E}_{x,y \sim p} [\ell_{bce}(C_\theta(x), y)] \) being the local loss and \( \mathcal{L}_{adv}(\theta) = \mathbb{E}_{x \sim p} [\log(1 - D_\phi(C_\theta(x))] \) the adversarial loss. For practical reasons of improved gradients behavior during training, it is common to substitute the symmetric formulation with \(-\mathbb{E}_{x \sim p} [\log D_\phi(C_\theta(x))]\).

During experimentation, we examined improved training stability and performance when using the Wasserstein-GAN with gradient penalty formulation (Arjovsky et al., 2017; Gulrajani et al., 2017). Thus, we allow \( D_\phi \) to output values in \( \mathbb{R} \) instead of just the segment \([0, 1] \), and replace the previous adversarial objective with:

\[
V(\theta, \phi) = \mathbb{E}_{y \sim p} [D_\phi(y)] - \mathbb{E}_{x \sim p} [D_\phi(C_\theta(x))] \quad (4)
\]

As customary with deep neural networks, stochastic gradient descent based optimizers are used for training. Each step, we perform a single update per model on their respective objectives. We use the same sampled batch for updating both models, as depicted in Algorithm 1. Sharing the batches enables reuse of the classifier outputs, reducing computational cost. In terms of performance, sampling different batches for the classifier and discriminator obtains similar results. Hence, we use shared batches.

3.2. Adversarial Loss as Prior Regularization

The auxiliary adversarial loss plays a central role in our method. Reaching an understanding of its effects is there-fore of utmost importance. While there are no currently existing general theoretical guarantees for convergence and optimality in deep models and adversarial methods, analyzing the objectives under such assumptions can lead to useful perspectives.

As shown in Goodfellow et al. 2014, assuming an infinite capacity and optimal discriminator, the original adversarial objective with respect to \( \theta \) is equivalent to the Jensen-Shannon divergence between \( p_{\theta} \) and \( p \). Under the same optimality assumptions, different adversarial losses produce objectives that are equivalent to alternative divergences. For example, the WGAN loss in use corresponds to the Earth Mover distance (Arjovsky et al., 2017). Therefore, the optimal \( \theta \) satisfies \( p_{\theta} = p \), obtaining the minimal divergence value. When this holds, \( D_\phi \) is unable to distinguish between the distributions, and \( C_\theta \) has fully captured the label distribution. In practice, however, both assumptions do not hold. Still, neural networks form a highly expressive function family, exhibiting impressive results when a right balance between the adversaries is attained.

From the above perspective, the multitask loss of the classifier is decomposed into two parts: a standard classification loss, and an adversarial regularization term. Minimization of the adversarial loss regularizes predictions towards the prior label distribution, reducing the likelihood of infeasible assignments. Hence, the \( \lambda_{adv} \) hyperparameter controls the trade-off between implicit structure learning, unconditioned on the input \( x \), and fitting the training data. In its extremities we obtain either a per label independent classifier when \( \lambda_{adv} = 0 \), or a model that attempts only to match the distribution \( p \) with \( p_{\theta} \), ignoring the input completely, when \( \lambda_{adv} \to \infty \).

3.3. True Label Smoothing

Joint training of adversarial networks relies on a delicate balance between the performance of the adversaries for stable and successful optimization. Practical experience has shown the benefits of several heuristics, mostly for image

---

**Algorithm 1 ALPR training procedure**

**Input:** training data \((X, Y)\), initial weights \((\theta, \phi)\), adversarial loss coefficient \(\lambda_{adv}\)

**for** number of training iterations **do**

Sample a batch of examples \(\{(x_i, y_i)\}_{i=1}^m\)

Perform discriminator descending step with gradient:

\[
\nabla_\phi \left[ \frac{1}{m} \sum_{i=1}^m [-D_\phi(y_i) + D_\phi(C_\theta(x_i))] \right]
\]

Perform classifier descending step with gradient:

\[
\nabla_\theta \left[ \frac{1}{m} \sum_{i=1}^m [\ell_{bce}(C_\theta(x_i), y_i) - \lambda_{adv} D_\phi(C_\theta(x_i))] \right]
\]

**end for**
We evaluate and compare two smoothing methods: Non-adaptive Smoothing and Adaptive Smoothing. Both methods perturb the true labels by adding small uniformly sampled noise. Formally, for a binary multivariate label $y = (y_1, ..., y_m)$ we perform the following update:

$$y_i \left\{ \begin{array}{ll} y_i + U[0, \beta_{neg}] & \text{if } y_i = 0 \\ y_i - U[0, \beta_{pos}] & \text{if } y_i = 1 \end{array} \right. \quad (5)$$

where $\beta_{neg}$ and $\beta_{pos}$ are noise margins that determine the size of the perturbation. The definition of these small noise margins is the differentiating factor between the methods.

Serving as a naïve baseline, Non-adaptive Smoothing uses predetermined constant values for $\beta_{neg}$ and $\beta_{pos}$. In contrast, Adaptive Smoothing employs an adaptive regime, updating the noise margins dynamically throughout training. Specifically, $\beta_{neg}$ and $\beta_{pos}$ are exponential moving averages of the classifier’s average prediction margins. A hyperparameter $\alpha \in [0, 1]$ controls the exponential decay rates of these moving averages, creating the following update:

$$\beta_{neg} \leftarrow \alpha \cdot \beta_{neg} + (1 - \alpha) \cdot \frac{\sum_{i=1}^{m} I[y_i < 0.5] \cdot \hat{y}_i}{\sum_{i=1}^{m} I[y_i < 0.5]}$$
$$\beta_{pos} \leftarrow \alpha \cdot \beta_{pos} + (1 - \alpha) \cdot \frac{\sum_{i=1}^{m} I[y_i \geq 0.5] \cdot (1 - \hat{y}_i)}{\sum_{i=1}^{m} I[y_i \geq 0.5]} \quad (6)$$

where $\hat{y}$ denotes a continuous label prediction for a specific input. These averages are updated after each forward pass of the classifier, and are also averaged over the whole batch. Averaging over the batch dimension is omitted from our notation for clarity purposes.

In Section 5.3 we evaluate the effects of the proposed smoothing methods on the resulting classifier and discriminator performance.

4. Related Work

Our work combines two relatively separate lines of work: deep structured prediction and adversarial learning. We briefly overview related studies, contrasting our method to existing approaches.

Undirected graphical models, known as Markov Random Fields (MRFs), and their discriminative variant Conditional Random Fields (CRFs) have been extensively studied in structured prediction literature (Lafferty et al., 2001; Taskar et al., 2004; Tschantzaridis et al., 2004). Typically, prediction is done by maximizing a factorized score with respect to the labels. To ensure tractable learning and inference, the factors are limited to small subsets or pairs with a simple graph structure (e.g., tree graph), not allowing to model higher order interactions. Increasing structure complexity is possible, however, it entails usage of approximation algorithms (e.g., Finley & Joachims, 2008; Kulesza & Pereira, 2008). Furthermore, in many cases (e.g., multi-label classification), the relevant dependencies are difficult to define through factors a priori.

There has recently been an increasing interest in usage of deep non-linear models for structured prediction. Initially, neural networks were introduced into existing graphical model methods in a two step procedure (Alvarez et al., 2012; Chen et al., 2014): First, pre-training deep models to create local potentials, typically replacing shallow linear functions. Then, producing the final score function by fixing the potentials and learning the interactions between them. These methods were later improved upon by unified approaches that combine both steps into a single end-to-end training procedure (Chen et al., 2015; Schwing & Urtasun, 2015; Zheng et al., 2015; Ma & Hovy, 2016). While achieving impressive results on several tasks, such as sequence tagging and image segmentation, the limitations on the possible graph structures remain. In contrast, our approach does not suffer from these restrictions, implicitly learning unconstrained structure over the labels.

Recent work of Belanger & McCallum 2016 has sparked interest in models that reduce the inductive bias posed by graphical models via implicitly learning structure in an unconstrained manner. Belanger & McCallum 2016 propose Structured Prediction Energy Networks (SPENs): deep neural networks that define a global energy function over the inputs and labels, relying on gradient-based inference for finding a minimizing prediction. Originally trained with an SSVM loss, subsequent work suggested alternative methods, such as end-to-end training (Belanger et al., 2017) and directly fitting the task loss (Gygli et al., 2017). Further work has also been done on generalizing SPENs by adding non-linear transformations on top of the score functions (Grabher et al., 2018), and on enhancing it with cardinality constraints (Brukhim & Globerson, 2018). Allowing unconstrained im-
Adversarial Loss as Prior Regularization for Structured Prediction

Table 1. Properties of the multi-label classification datasets.

<table>
<thead>
<tr>
<th>DATASET</th>
<th>#FEATURES</th>
<th>#LABELS</th>
<th>#TRAIN</th>
<th>#TEST</th>
<th>AVG LABELS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bibtex</td>
<td>1836</td>
<td>159</td>
<td>4880</td>
<td>2515</td>
<td>2.40</td>
</tr>
<tr>
<td>Bookmarks</td>
<td>2150</td>
<td>208</td>
<td>60000</td>
<td>27856</td>
<td>2.03</td>
</tr>
<tr>
<td>Delicious</td>
<td>500</td>
<td>983</td>
<td>12920</td>
<td>3185</td>
<td>19.02</td>
</tr>
</tbody>
</table>

Explicit structure learning lead to impressive results in several structured prediction tasks. Inspired by these approaches, our method enables explicit learning of unconstrained dependencies for simple classifiers through an adversarial loss, improving results and substantially decreasing inference complexity.

On a mostly separate track of research, adversarial learning has been successfully used to train likelihood-free generative models for sampling from high-dimensional distributions. Since the introduction of generative adversarial networks (GANs) in the seminal work of Goodfellow et al. (2014), numerous studies have been dedicated to improving training stability and generation capabilities through different objectives and architectures (e.g., Radford et al., 2015; Salimans et al., 2016; Arjovsky et al., 2017; Gulrajani et al., 2017). Motivated by the impressive results for image generation, we utilize an adversarial loss for regularizing predictions towards the true joint label distribution.

Perhaps most related to our work, several recent papers propose usage of an adversarial framework for structured prediction tasks. Aiming to speed-up inference time for SPENs, Tu & Gimpel (2018) propose replacing gradient-based inference with an inference network that predicts the minimizing label assignment of the energy function. The training procedure optimizes the energy and inference networks in an alternating fashion, resulting in a formulation that resembles GANs (Goodfellow et al., 2014). Inference networks greatly simplify inference of SPENs, reducing prediction time significantly. However, in terms of performance they do not show substantial improvements over SPENs.

On semantic image segmentation and depth estimation tasks, adversarial learning has been used for enforcing prediction consistencies, improving performance over several baselines (Luc et al., 2016; Hwang et al., 2019). Finally, Ren et al. (2018) use adversarial learning as a means to implicitly learn constraints from data. They present a semi-supervised method that learns from labels with no corresponding inputs, achieving impressive results on a number of regression tasks with limited or no labeled data.

Our approach differs in several key points. We focus on an unconditional discriminator setup that receives only the predicted or actual labels, without any knowledge of the corresponding inputs. Moreover, we show the applicability of an adversarial framework to a different domain, improving existing results on multi-label classification benchmarks.

5. Experiments

We evaluate our approach on binary multi-label classification datasets used in recent work, and compare it to alternative methods (Lin et al., 2014; Belanger & McCallum, 2016; Gygli et al., 2017; Tu & Gimpel, 2018; Brukhim & Globerson, 2018). Multi-label classification poses an interesting challenge for models that learn structure implicitly, such as our own, since it contains correlations between labels that are hard to define a priori. All 3 datasets used: Bibtex, Bookmarks (Katakis et al., 2008), and Delicious (Tsoumakas et al., 2008) contain binary bag-of-words representations of textual inputs and their corresponding binary tags. The textual inputs are extracted from metadata of bibtext items, web browser bookmarks, and contents of web pages, respectively. Dataset sizes and other properties are presented in Table 1.

5.1. Experimental Setup

In what follows we describe the experimental setup, including our model architecture, hyperparameters, training procedure, and evaluated methods. The hyperparameters were tuned on a development set that was cut out from the training set using a 80%-20% split. Then, similar to alternative methods, we train the final model on the entire training set before evaluating on the test set. Unless stated otherwise, the following setup is identical for all datasets.

Architecture: Both the classifier and the discriminator are multilayer perceptrons with one hidden layer of size 150 and ReLU activations.

Regularization: Empirically, we found regularization to play a key role in performance on the evaluated benchmarks, especially for the relatively small Bibtex dataset. We apply dropout (Srivastava et al., 2014) with $p = 0.5$ to the hidden layers of the classifier and discriminator. Additionally, we use $L_2$ regularization with coefficient $\lambda_{L_2} = 10^{-5}$ on all datasets for the discriminator, and only on Bibtex for the classifier. Furthermore, as pointed out in Section 3.2, the adversarial loss can also be seen as a form of regularization.

Training: We trained the model with a batch size of 64 for 30 epochs on the Bibtex and Delicious datasets, and
only 10 for Bookmarks due to its size. We examined that performance on the development set saturates and does not improve when increasing the number of training epochs. For optimization, we used the Adam optimizer (Kingma & Ba, 2014) with learning rates of $10^{-3}$ and $10^{-4}$ for the classifier and discriminator, respectively. Lastly, the adversarial loss coefficients were set to the values $\lambda_{adv} = 0.15, 0.05, 0.005$ for Bibtex, Bookmarks and Delicious, respectively. The discriminator is optimized with a lower learning rate to help balance between the adversaries. Experimentally, we have found that limiting the discriminator improved overall performance and reduced sensitivity to changes in $\lambda_{adv}$.

Additional hyperparameters are the adaptive smoothing decay rate $\alpha$ which we set to 0.9, the WGAN gradient penalty coefficient which we set to 5, and finally, the classification thresholds. For Bibtex and Bookmarks the chosen threshold was 0.1, while for Delicious we used 0.15.

**Evaluated Methods:** We evaluate and compare the following methods:

- **PRLR** (Lin et al., 2014): An approach specifically for sparse multi-label classification tasks. Proposes a posterior regularization technique for encouraging sparse and low-rank predictions.

- **MLP (SPEN)** (Belanger & McCallum, 2016): A multilayer perceptron baseline used in the original SPEN paper. This model has 2 hidden layers of size 50 and uses ReLU non-linearities.

- **SPEN** (Belanger & McCallum, 2016): Structured Prediction Energy Networks define an energy function as a sum of local and global potentials, using gradient-based inference.

- **DVN** (Gygli et al., 2017): Deep Value Networks use the same architecture and inference as SPENs, altering the training objective from an SSVM loss to directly fitting a task loss.

- **InfNet** (Tu & Gimpel, 2018): Inference Networks replace gradient-based inference with a network that outputs an approximation of the optimal labels for SPENs.

- **P&C** (Brukhim & Globerson, 2018): Predict and Constrain incorporates cardinality constraints and end-to-end unrolled optimization into a SPEN based model.

- **MLP (Ours):** Our multilayer perceptron baseline with one hidden layer of size 150, dropout, and ReLU activations. This model is equivalent to ALPR when setting $\lambda_{adv}$ to 0.

- **ALPR:** Our adversarial approach with adaptive smoothing for the true label inputs of the discriminator, as described in Section 3.3.

<table>
<thead>
<tr>
<th>Method</th>
<th>BibTex</th>
<th>Bookmarks</th>
<th>Delicious</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRLR</td>
<td>44.2</td>
<td>34.9</td>
<td>33.3</td>
</tr>
<tr>
<td>MLP (SPEN)</td>
<td>38.9</td>
<td>33.8</td>
<td>37.8</td>
</tr>
<tr>
<td>SPEN</td>
<td>42.2</td>
<td>34.4</td>
<td>37.5</td>
</tr>
<tr>
<td>DVN</td>
<td>44.7</td>
<td>37.1</td>
<td>-</td>
</tr>
<tr>
<td>InfNet</td>
<td>42.2</td>
<td>37.6</td>
<td>37.5</td>
</tr>
<tr>
<td>P&amp;C</td>
<td>45.5</td>
<td>39.1</td>
<td><strong>38.4</strong></td>
</tr>
<tr>
<td>MLP (Ours)</td>
<td><strong>44.6</strong></td>
<td>37.9</td>
<td>37.7</td>
</tr>
<tr>
<td>ALPR</td>
<td><strong>47.2</strong></td>
<td><strong>39.8</strong></td>
<td>37.7</td>
</tr>
</tbody>
</table>

For fair comparison, our network architecture is similar to the architectures used in the methods above, and we follow the same evaluation protocol. Specifically, we use the same train-test splits and report the example averaged $F_1$ score.

### 5.2. Results

As can be seen from the results in Table 2, ALPR outperforms existing approaches on the Bibtex and Bookmarks datasets, achieving absolute improvements over the previous top performing method of 1.7% and 0.7%, respectively. Moreover, ALPR substantially improves upon the results of a standard MLP classifier on these datasets. Since the only differentiating factor is the additional adversarial loss, these results demonstrate its effectiveness in improving performance of standard classifiers by considering global structure. In terms of inference cost, our method greatly simplifies the prediction process compared to the iterative gradient-based inference in the SPEN, DVN, and P&C approaches. When comparing to the PRLR, InfNet, and MLP baselines, which similarly to our method require only a single forward pass for inference, we observe an even larger margin in performance.

On the contrary, on the Delicious dataset the auxiliary adversarial loss does not improve upon a standard MLP. Excluding P&C, existing methods struggle on this dataset, falling behind simple per label independent baselines. We attribute this lack of improvement to the unique label properties of the dataset, which contains more labels than features and has a significantly larger average label cardinality compared to Bibtex and Bookmarks (Table 1). It seems that implicit structure learning models are not able to fully capture existing label dependencies in such a case, as opposed to P&C which explicitly considers cardinality constraints.

Interestingly, our MLP classifier reaches competitive results, surpassing several structured methods and exhibiting substantial improvements over a similar baseline previously.
Adversarial Loss as Prior Regularization for Structured Prediction

Figure 2. Train and test $F_1$ scores versus the adversarial loss coefficient, $\lambda_{adv}$, on the Bibtex dataset.

reported by Belanger & McCallum 2016. The MLP baselines differ in architecture and regularization. Our baseline contains a single hidden layer and a dropout layer, while Belanger & McCallum 2016 opted for two hidden layers. The performance difference suggests that regularization is a crucial factor for generalization on these datasets, perhaps explaining in part ALPR’s success.

Figure 2 further showcases the regularization properties of the adversarial loss, depicting a standard relation between performance and strength of regularization. As the adversarial loss coefficient increases, we initially observe an increase in test set performance, while the train set performance decreases. Then, larger coefficient values cause an overall degradation, overly impairing the model.

5.3. Ablation Study

In this section we analyze the effects of the smoothing methods proposed in Section 3.3. Additionally, we compare them to a No Smoothing configuration, which uses the original binary labels without any smoothing. We note that the Adaptive Smoothing configuration is the same ALPR model reported in Table 2, using $\alpha = 0.9$ decay rate. For Non-adaptive Smoothing, we tuned the constant noise margin values on the development set, and report the results for $\beta_{neg}, \beta_{pos} = 0.05$.

As can be seen in Table 3, our proposed adaptive smoothing method outperforms the naïve non-adaptive baseline. Static noise margins are not able to mitigate the representational discrepancy, resulting in an overly accurate discriminator that prevents meaningful learning through the adversarial objective. It is interesting to notice that the Adaptive Smoothing and No Smoothing methods reach comparable results.

Table 3. Test set $F_1$ scores of our proposed ALPR model with different discriminator input smoothing methods. The reported smoothing methods are: Adaptive Smoothing (AS), Non-adaptive Smoothing (NAS), and No Smoothing (NS).

<table>
<thead>
<tr>
<th>CONFIGURATION</th>
<th>BIBTEX</th>
<th>BOOKMARKS</th>
<th>DELICIOUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPR - AS</td>
<td>47.2</td>
<td>39.8</td>
<td>37.7</td>
</tr>
<tr>
<td>ALPR - NAS</td>
<td>45.2</td>
<td>39.3</td>
<td>37.6</td>
</tr>
<tr>
<td>ALPR - NS</td>
<td>47.0</td>
<td>39.7</td>
<td>37.7</td>
</tr>
</tbody>
</table>

Despite the possible trivial solutions for separating between binary true labels and the classifier’s predictions, we observe similar performance gains over the MLP baseline in both cases. Examining the discriminator accuracy obtained on the test sets reveals that without perturbing the true label inputs it reaches extremely high values of over $\sim 99.5\%$. Usage of the non-adaptive smoothing approach results in a similar outcome. Meanwhile, adaptively smoothing the true label inputs of the discriminator succeeds in slightly degrading the discriminator’s accuracy to $\sim 94\%$, however, the reduced accuracy does not translate to improved classifier performance. This behavior is consistent across all 3 datasets, leaving room for further investigation of smoothing techniques.

6. Conclusion and Future Work

We present a flexible adversarial learning framework for deep structured prediction that incorporates an auxiliary adversarial loss for training a classification model. The adversary enhances the classifier with implicit structure learning capabilities by regularizing predictions towards matching the prior joint label distribution. As a result, we obtain state-of-the-art performance, surpassing currently existing methods on multi-label classification benchmarks while simplifying inference.

Future work can explore usage of ALPR for other domains with complex structured outputs. Introducing an adversarial loss on top of existing methods often does not require any further modifications. However, attaining stable joint training in adversarial setups remains a difficult challenge. Another important direction is exploring the implicit structure learning capabilities of recent deep models. While achieving impressive results, the learned dependencies are still a mystery, for the most part. A better understanding may be key for making meaningful progress in structured prediction.

References

Alvarez, J. M., LeCun, Y., Gevers, T., and Lopez, A. M. Semantic road segmentation via multi-scale ensembles of


