Implicit Training of Inference Network Models for Structured Prediction

Shiv Shankar[1]

[1]University of Massachusetts, USA

Abstract

Most research in deep learning has predominantly focused on the development of new models and training procedures. In contrast, the exploration of training objectives has received considerably less attention, often limited to combinations of standard losses. When dealing with complex structured outputs, the effectiveness of conventional objectives as proxies for the true objective becomes can be questionable. In this study, we propose that existing inference network-based methods for structured prediction, as observed in previous works [Tu and Gimpel, 2018, Tu et al., 2020a], indirectly learn to optimize a dynamic loss objective parameterized by the energy model. Based on this insight, we propose a method that treats the energy network as a trainable loss function and and employ implicit-gradient-based technique to learn the corresponding dynamic objective. We experiment with multiple tasks such as multi-label classification, entity recognition etc. and find significant performance improvements over baseline approaches. Our results demonstrate that implicitly learning a dynamic loss landscape proves to be an effective approach for enhancing model performance in structured prediction tasks.

1 INTRODUCTION

Deep neural networks have achieved widespread success in a multitude of applications such as translation [Vaswani et al., 2017], image recognition [He et al., 2016] and many others. This success has been enabled by the development of back-propagation based algorithms, which provide a simple and effective way to optimize a loss calculated on the training set. Generally a large portion of existing work has focused only on designing of models and optimization algorithms. However with the increased prevalence of meta-learning, researchers are exploring new loss objectives and training algorithms [Wu et al., 2018, Huang et al., 2019].

Intuitively one would like to choose objectives which can dynamically refine the kind of signals it produces for a model to follow, in order to guide the model towards a better solution. Oftentimes standard objectives are pretty effective at this; however, these objectives have generally been explored for simple predictions. When dealing with complex outputs, there is a significant scope for improvement by designing better training objectives. A good example is structured prediction [Belanger and McCallum, 2016], where the output includes multiple variables and it is important to model their mutual dependence. One natural candidate is to use the likelihood under a probabilistic model that captures this dependence. Such models though cannot be used to efficiently predict the output and require inference.

An ideal loss function in this case would naturally guide the model towards incorporating the output correlations while allowing a more standard feed-forward or similar predictive model to quickly and efficiently produce the output. Energy based structured prediction [Belanger and McCallum, 2016] provide a natural framework in which one can explore learned losses by using the energy itself as the training objective. Existing works [Tu and Gimpel, 2018, Tu et al., 2020a] have looked at learning prediction networks to directly predict structured outputs, and not on the energy-based objective itself.

Contributions This work explores the thread of learning dynamic objectives for structured prediction. Using the insight of Hazan et al. [2010], we connect the existing paradigm of Tu et al. [2020a], Lee et al. [2021] to a surrogate loss learning problem. This allows us to identify a key problem with the approach of Tu et al. [2020a], Lee et al. [2021], that it uses incorrect gradient for the surrogate objective problem. Building on this idea, we propose to use implicit gradients [Krantz and Parks, 2002] for learning an energy based structured prediction model. We then use ideas from

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

Christianson [1992] to compute gradients at scale for the corresponding optimization problem. The experimental results show the effectiveness of our methods against SOTA baselines on three tasks and nine datasets.

2 PRELIMINARIES

2.1 LEARNING A LOSS FUNCTION

We describe in this section a formulation for learning a dynamic loss function. This loss function, which we call auxiliary loss is used to train a *model*. The model trained on this auxiliary loss is then evaluated on a different loss function, called the primary loss. This second loss function is called primary loss because this is the 'true' loss of concern to the user. For example, in a standard supervised learning setting, the primary loss could be the performance of the model on a validation set. The goal then is to learn the auxiliary loss in a way that the learnt model's performance as measured by the primary loss is optimized. If the model is denoted by f_θ with parameters θ, auxiliary loss by $\mathcal{L}_{\text{Aux}}$, and primary loss by $\mathcal{L}_{\text{Prim}}$; then this problem can be written as:

$$\min_{\mathcal{L}_{\text{Aux}}} \mathcal{L}_{\text{Prim}}(\operatorname*{argmin}_{\theta} \mathcal{L}_{\text{Aux}}(\theta)) \tag{1}$$

Variants of the same formulation have been explored for supervised learning [Huang et al., 2019, Wu et al., 2018] and reward learning [Bechtle et al., 2019, Zheng et al., 2018]. The outer problem is technically a problem of optimization over the space of functions. To be able to solve this computationally, the auxiliary loss $\mathcal{L}_{\text{Aux}}$ is often parameterized with some parameters ϕ; changing the problem to learning ϕ.

$$\begin{aligned} &\phi^* = \operatorname*{argmin}_{\phi} \mathcal{L}_{\text{Prim}}(\theta^*(\phi), \phi) \\ &\text{such that} \\ &\theta^*(\phi) = \operatorname*{argmin}_{\theta} \mathcal{L}_{\text{Aux}}(\theta, \phi) \end{aligned} \tag{2}$$

2.2 IMPLICIT GRADIENT METHOD

The aforementioned problem is a bi-level optimization problem. In such a case, a parameter (ϕ) that influences $\mathcal{L}_{\text{Aux}}$, can influence the primary objective $\mathcal{L}_{\text{Prim}}$ via the dependence of the inner optimized parameters $\theta*$ on ϕ. The implicit gradient method [Krantz and Parks, 2002, Dontchev and Rockafellar, 2009] provides a way to compute the gradient of $\mathcal{L}_{\text{Prim}}$ wrt ϕ due to this implicit dependence.

For the problem given in Equation 2, under certain regularity conditions, $\mathcal{L}_{\text{Prim}}$ is a differentiable function of ϕ and its gradient is given by:

$$\frac{\partial \mathcal{L}_{\text{Prim}}}{\partial \phi} = \underbrace{\frac{\partial(\mathcal{L}_{\text{Prim}}(\theta^*(\phi), \phi))}{\partial \phi}}_{\text{Explicit gradient}} - \underbrace{\left[\frac{\partial}{\partial \phi}\frac{\partial}{\partial \theta}(\mathcal{L}_{\text{Aux}}(\theta, \phi))\right]\left[\frac{\partial}{\partial \theta}\frac{\partial}{\partial \theta}(\mathcal{L}_{\text{Aux}}(\theta, \phi))\right]^{-1}\frac{\partial(\mathcal{L}_{\text{Prim}}(\theta^*(\phi), \phi)}{\partial \theta}}_{\text{Implicit Gradient}} \tag{3}$$

The existence of gradient follows from Theorem 2G.9 in Dontchev and Rockafellar [2009]. The derivation of the Equation 3 is presented in the Appendix. As can be seen from the above equation, the true gradient has two terms: a standard component $\partial_\phi \mathcal{L}_{\text{Prim}}$ and the implicit component due to the the dependence of optimal θ on ϕ. We will sometimes abuse terminology to call this term as implicit or meta gradient.

3 STRUCTURED PREDICTION WITH DYNAMIC LOSS

In this section we provide a brief overview of structured prediction, before we present a bi-level optimization based method for structured prediction. Unlike standard classification, structured prediction deals with predicting a multivariate structured output such as multi-label outputs, semantic labeling etc.

In this work we consider a structured prediction task as learning a mapping from an input space $\mathcal{X}$ to a exponentially large label space: $\mathcal{Y}$. The quality of a predicted output is determined by the score function $s : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. The score function is used to compare the gold output $y \in \mathcal{Y}$ with another output $y' \in \mathcal{Y}$ and can be interpreted as a measure of how good y' is compared to y. Some common scoring functions are BLEU used for translation [Papineni et al., 2002], Hamming Distance for comparing strings, and F1-score for multilabel classification tasks [Kong et al., 2011].

While in principle, a powerful enough neural network can learn to directly predict the structured output; in practice they often not perform well due to a) ignoring the dependencies between the labels and b) not using or optimizing for the task loss.

Energy Based Structured Prediction Energy-based structured prediction aims to alleviate these challenges by proposing to learn an energy network $E_\phi : \mathcal{X} \times \bar{\mathcal{Y}} \to \mathbb{R}$ which provides the energy for pairs of inputs x and the outputs y. Here $\bar{\mathcal{Y}}$ refers to a suitable relaxation of $\mathcal{Y} \in \{0, 1\}^L$ to a continuous space: $\bar{\mathcal{Y}} \in [0, 1]^L$. The energy network is trained to assign the correct output y lower energy than incorrect outputs. At test time, predictions are recovered for an input x by finding the structure y with the lowest energy [Belanger and McCallum, 2016]. Training energy

models this way requires inference during training to produce high scoring negative samples negative $\bar{y}$. With such negative samples, a common practice is to use SSVM loss [Tsochantaridis et al., 2004].

$$\min_{\phi} \underbrace{[s(\bar{y}, y) - E_{\phi}(x, \bar{y} + E_{\phi}(x, y)]_{+}}_{I} \tag{4}$$

SPEN [Belanger and McCallum, 2016] propose using gradient based inference to find $\bar{y}$, which makes training and inference slow. To make this eff Tu and Gimpel [2018], Tu et al. [2020a] propose using *inference networks* F_{θ} and A_{θ} to directly predict the output. F_{θ} is the cost-augmented inference network that is used only during training i.e. its output is used as $\bar{y}$ in 4. As such F_{θ} is trained to maximize 4 with respect to $\bar{y}$. On the other hand, the goal of A_{θ} is to predict the output during testing, and as such A_{θ} is trained to minimize the energy.

3.1 USING ENERGY AS PROXY LOSS

An ideal predictor would be directly minimizing the structured loss s. However, due to the nature of many real-life structured losses (like BLEU, F1, IoU), and due to the often discrete nature of the output space, performing such an inference is intractable. A natural alternative then is using deep networks to build a proxy or surrogate loss. In fact, the classic cost sensitive hinge/margin loss used in Tsochantaridis et al. [2004] (i.e. 4) is a convex surrogate of the true cost [Hazan et al., 2010]. Similarly the value network method of Gygli et al. [2017] aims to learn a differentiable energy network which directly predicts the score/task-loss of an output. This suggests that the training of the energy network in energy-based structured prediction methods an indirect way to learn a surrogate loss function. This can also be seen by the fact that the energy function E appears additively in the training objective for the inference net A.

Surrogate loss learning can be formulated as a bi-level optimization with the outer optimization over loss function parameters ϕ constantly updating itself to provide better feedback to the prediction model (as discussed in Section 2). Under this view the margin loss based training can be interpreted as the following bi-level objective:

$$\min_{\phi} \mathcal{L}_{\text{Prim}}(\theta(\phi), \phi) \quad \text{s.t.} \quad \theta(\phi) = \operatorname*{argmin}_{\theta} \mathcal{L}_{\text{Aux}}(\phi, \theta)$$

where

$$\mathcal{L}_{\text{Aux}}(\theta, \phi) = -(s(F_{\theta}(x), y) + E_{\phi}(x, F_{\theta}(x)) + \lambda E_{\phi}(x, A_{\theta}(x)))$$

and

$$\mathcal{L}_{\text{Prim}}(\theta, \phi) = [s(F_{\theta}(x), y) - E_{\phi}(x, F_{\theta}(x)) + E_{\phi}(x, y)]_{+} + \lambda[-E_{\phi}(x, A_{\theta}(x) + E_{\phi}(x, y)]_{+}$$

The interpretation of the training procedure is that at each step, the inference network is trained to predict an incorrect $\bar{y}$ with low energy and then the energy network is updated to guide the inference network to a newer solution. Next we note that under a well trained E one does not need two different networks A, F, and so we combine the two of them in the same network. To train this model, we use gradient descent based optimization, however, instead of backpropagating through the gradient steps, we use the implicit gradient method to obtain the gradients of ϕ.

Under our interpretation, the procedure of Tu et al. [2020a], Lee et al. [2021] uses biased gradients during update of ϕ. Specifically since they only use $\partial_{\phi}\mathcal{L}_{\text{Prim}}(\theta, \phi)$, their gradient for ϕ misses the second term (labeled implicit gradient) in Equation 3, which captures the influence due to the implicit dependence of θ on ϕ. Specifically the presence of the mixed derivatives serve the purpose of mapping changes in ϕ and θ into each other. The presence of the inverse Hessian in the missing term provides insight into why the bi-level approach can be superior. Note that the condition number of the Hessian is a useful measure of the hardness of an optimization and an ill-conditioned Hessian would cause the missing term to explode, something which the adversarial training process of Tu et al. [2020a] ignores. We present a more detailed discussion of this in the Appendix.

One can observe that in the above optimization for θ, the observed outputs y only appear directly in the score function s. If s is not differentiable (which is usually the case), updates to A_{θ} relies on the function E. However, during initial steps of training, E_{ϕ} would not yet have learned to score the true output correctly. Thus the model A_{θ} will receive poor supervision. To alleviate this issue, we add direct supervision from output y in $\mathcal{L}_{\text{Aux}}$. In this case we use the output of A_{θ} to construct a distribution which is updated via the cross entropy (CE)/ log-likelihood (MLE) loss.

While one can use different parameters θ, θ' to parameterize the F, A networks respectively, for a well trained energy model these networks are not very dissimilar in behaviour. Furthermore, Tu et al. [2020a] also found sharing parameters between F and A helpful. Hence we also consider these as the same network. Putting these changes (i.e. merging of inference networks and addition of MLE loss) together we get the following auxiliary objective:

$$\mathcal{L}_{\text{Aux}} = \mathcal{L}_{MLE}(y, A_{\theta}(x)) + \lambda E_{\phi}(x, A_{\theta}(x)) \tag{5}$$

where $\mathcal{L}_{MLE}$ is a log-likelihood/cross-entropy based loss and λ is a hyperparameter.[1]

Remark 1. Unlike standard meta-learning problems, where the outer parameter ϕ is used as the initialization point of

[1] If we replace F by A in $\mathcal{L}_{\text{Aux}}$ objective above both energy based terms become same; next since s is not-directly optimizable we replace it for supervision with $\mathcal{L}_{MLE}$

the model, here we can directly use the learned inference network A_θ for prediction. However, one can refine the final θ on a validation set, or attempt to refine the output of network $A_\theta(x)$ via gradient descent on the energy E_ϕ. We do not use the validation set for further refinement in our experiments.

Remark 2. While ideally the inference networks would be predicting binary vectors, in practice they are used to predict soft examples. The energy function can then be updated both on the real valued vectors or by obtaining hard outputs via sampling, rounding or other methods.

3.2 SCALABLE COMPUTATION OF THE IMPLICIT-GRADIENT

An astute reader might note that computing the gradient given in Equation 3 directly requires the Hessians $\frac{\partial}{\partial\theta}\frac{\partial}{\partial\theta}(\mathcal{L}_{\text{Aux}}(\theta,\phi))$ and $\frac{\partial}{\partial\phi}\frac{\partial}{\partial\theta}(\mathcal{L}_{\text{Aux}}(\theta,\phi))$. While computing the Hessians can be compute-intensive if the dimensionality of the parameters θ,ϕ is large; computing the inverse Hessian is prohibitively more so. An alternative method is to differentiate through the optimization procedure, however that severely limits the number of optimization steps one can conduct. Moreover truncated optimization will induce its own biases [Vollmer et al., 2016].

Fortunately, we do not need to compute any of the two matrices. Instead we only need the vector product of these hessian matrices (HVP) with the gradient $\frac{\partial(\mathcal{L}_{\text{Prim}}(\theta^*(\phi),\phi)}{\partial\theta}$. Efficiently doing such operations is a well researched area with numerous methods [Christianson, 1992, Vázquez et al., 2011, Song and Vicente, 2022]. The given expression can be transformed into first computing a HVP with the cross-Hessian $\frac{\partial}{\partial\phi}\frac{\partial}{\partial\theta}(\mathcal{L}_{\text{Aux}}(\theta,\phi))$, and then into an inverse-Hessian vector product (iHVP) with the Hessian $\frac{\partial}{\partial\theta}\frac{\partial}{\partial\theta}(\mathcal{L}_{\text{Aux}}(\theta,\phi))$. For the inverse Hessian, we use the von-Neumann expansion method suggested in Lorraine et al. [2020]. This allows one to convert iHVP with a matrix H to product to a polynomial in HVP using the same matrix H (details in the Appendix). Once every requisite operation has been turned to HVP, we can use auto-differentiation on perturbed parameters (i.e. finite step divided difference approximation).

3.3 PRIMARY LOSS DESIGN

An advantage of breaking this problem as a bi-level optimization is that unlike [Tu et al., 2020a] where the objectives being used for training ϕ,θ are by construction adversarial, we can now use different objectives for our primary and auxiliary losses. We implicitly already used this fact when we added the binary cross entropy loss to $\mathcal{L}_{\text{Aux}}$, and wrote slightly different form for $\mathcal{L}_{\text{Aux}}$ in Equation 5. However we also have the freedom to choose the primary loss $\mathcal{L}_{\text{Prim}}$ which can result in different behaviour for the models. In fact structured prediction literature has explored variety of losses for training energy models. We mention a few of these which we work use as $\mathcal{L}_{\text{Prim}}$ in our experiments. Some of these have also been explored for structured prediction by Lee et al. [2021]. In this section we shall often use $\bar{y}$ to denote an element from $\mathcal{Y}$ which is distinct from the true output y.

Hinge/SSVM Loss. Early structured prediction models were often trained with a version of the hinge loss adjusted for the score function [Tsochantaridis et al., 2004]. In current parlance it is also known as margin loss. This is one of the components of the loss used in [Tu et al., 2020a]. It is given by the following equation:

$$\mathcal{L}_{\text{SSVM}} = [s(\bar{y},y) - E_\phi(x,\bar{y}) + E_\phi(x,y)]_+$$

DVN Loss. A natural candidate for learning energy network is to use it as a differentiable proxy for the score function s. This can be done by match the energy values to the task loss i.e. $E(x,\bar{y}) \approx s(\bar{y},y)$. [Gygli et al., 2017] proposed to normalize the score function s and use a cross entropy loss to match the energy E with it.

$$\begin{aligned}\mathcal{L}_{\text{DVN}} = &- s(\bar{y},y)\log(-E_\theta(x,\bar{y})) \\ &- (1-s(\bar{y},y))\log(1+E_\theta(x,\bar{y}))\end{aligned}$$

Contrastive Divergence. Literature in probabilistic inference have proposed various losses to do maximum likelihood estimate of energy models [Gutmann and Hyvärinen, 2010]. A common loss for such training is the contrastive-divergence [Hyvärinen and Dayan, 2005] based loss which uses samples to approximate the log-likelihood of the model. We use a similar loss augmented with the score function s as shown below.

$$\mathcal{L}_{CD} = \log\frac{\exp(-E_\phi(x,y))}{\sum_{k=0}^{K}\exp(-E_\phi(x,\bar{y}_k)+s(\bar{y}_k,y)))}$$

where $\bar{y}_{1..K}$ refers to K possible negative (non-true output) samples and $\bar{y}_0 = y$.

Noise-Contrastive Loss. We also experiment with a modified version of the $\mathcal{L}_{CD}$ loss above which inspires from noise contrastive estimation [Ma and Collins, 2018]. Lee et al. [2021] have also used this version to train structured energy networks.

$$\mathcal{L}_{NCE} = \log\frac{\exp(-E_\phi(x,y)-\log P(y|x))}{\sum_{k=0}^{K}\exp(-E_\phi(x,\bar{y}_k)+s(\bar{y}_k,y))-\log P(\bar{y}_k|x))}$$

where $\bar{y}_{1..K}$ once again refers to K possible negative (non-true output) samples and $\bar{y}_0 = y$. $P(\bar{y}_k|x)$ is the probability of the value $\bar{y}_k$ as estimated by the predictive inference net

under the assumption that its components are independent i.e. $P(\bar{y}_k|x) = \prod_i P_\phi(\bar{y}_k^i|x)$

During training $\bar{y}$ in the aforementioned objectives gets replaced by the prediction of the inference net $A_\theta(x)$. When multiple values are required (such as for $\mathcal{L}_{CD}$) we obtain them samples by interpreting the continuous output of $A_\theta(x)_j$ as a Bernoulli random variables, and drawing samples from the corresponding distribution.

Remark 3. Learnt loss functions have been used in literature for the outer objective [Bechtle et al., 2019]. However, these are also loss objectives used to train the prediction model ($\mathcal{L}_{\text{Aux}}$ in our notation). In this work, predictions are obtained from the inference network A_θ, which is trained by optimizing the energy function E. Hence we call E dynamic loss in the latter sense.

Now we are in a position to state our exact proposal to train structured prediction models. Our proposed method is summarized in Algorithm 1. The network E is trained by an energy-learning-based objective to learn a landscape that incorporates signal from the task loss and reflects the dependencies among output variables. An energy optimum is indicative of a good prediction satisfying high similarity with the true output while respecting statistical dependence between labels. The inference net gets trained to predict an optima of the energy E. The algorithm updates the inference networks in the direction of reducing energy and the energy serves as a surrogate loss.

Algorithm 1 Implicit Gradient for structured prediction

Require: Energy Network E_ϕ, Inference Network A_θ,
  Regularization λ, Training Data $\mathcal{D} = x_i, y_i$,
  Inner/Outer Iterations $T_{\text{inner}}, T_{\text{outer}}$
  Sample θ_0, ϕ_0 randomly
  for $t \in T_{\text{outer}}$ iterations **do**
    Obtain sample x, y from $\mathcal{D}$
    $\theta_p \leftarrow \theta_t$
    for $p \in T_{\text{inner}}$ iterations **do**
      Compute $\mathcal{L}_{\text{Aux}}(\theta_p, \phi_t)$
      $\theta_p \leftarrow \theta_p - \eta \nabla \mathcal{L}_{\text{Aux}}(\theta_p, \phi_t)$
    end for
    $\theta_t \leftarrow \theta_p$
    Compute $\mathcal{L}_{\text{Prim}}(\theta_t, \phi_t)$
    Compute $g = \frac{d}{d\phi}\mathcal{L}_{\text{Prim}}(\theta_t, \phi_t)$ via Equation 3
    $\phi_{t+1} \leftarrow \phi_t - \eta g$
    $\theta_{t+1} \leftarrow \theta_t$
  end for
  Return resulting model A_θ

3.4 CONNECTION TO DYNAMIC LOSSES

A key distinction between our proposed method, employing the $\mathcal{L}_{SSVM}$ loss, and the InfNet+ approach suggested by Tu et al. [2020a] lies in the utilization of the implicit gradients/bi-level optimization method instead of alternative independent optimization. It is worth noting that when training via independent optimization, the parameters ϕ are updated on $\partial\phi\mathcal{L}_{\text{Prim}}$. However, Equation 3 reveals that $\partial_\phi\mathcal{L}_{\text{Prim}}$ represents only one component of the true gradient, as it incorporates an additional term (Line 1 of Equation 3). This additional term captures the indirect influence of θ on $\mathcal{L}_{\text{Prim}}$, as θ is also dependent on ϕ.

As such standard energy based training methods, use biased gradients which can lead to improper solutions to the underlying problem. However, by utilizing the correct gradient, the energy optimization (ϕ) becomes explicitly aware of and receives feedback from the inner optimization process of the inference network. To emphasize th distinction this causes from the standard training procedure of energy-based models (EBMs) described by Tu et al. [2020a], consider a scenario where the energy parameters E have fully optimized $\mathcal{L}_{\text{Prim}}$ (e.g., through margin loss). Suppose that at the current parameters θ of the inference net A, the energy of the output $E(x, A(x))$ has a singular hessian $\partial_\theta^2 E_\phi(x, A_\theta(x))$. Since the condition number of the Hessian often serves as a reliable indicator of the difficulty of convex optimization (e.g., gradient descent-based optimization becomes slow in the presence of a flat surface), it is reasonable to infer that finding the optimal y is challenging within the current energy landscape.

Under the standard EBM training, as the updates solely depend on $\partial_\phi\mathcal{L}_{\text{Prim}}$, the energy network remains oblivious to the difficulty associated with predicting y. Consequently, when $\mathcal{L}_{\text{Prim}}$ approaches optimality, the energy network ceases to receive updates. Conversely, in our framework, the update to the energy function is not solely determined by $\partial_\phi\mathcal{L}_{\text{Prim}}$. It is important to note that the first term in Equation 3 relies on the inverse Hessian of the loss with respect to θ. Therefore, if the Hessian of the inference network exhibits ill-conditioning, a significant gradient for the energy network emerges. This enables the energy function to discover a landscape that is adapted to the behavior of the inference network.

In this sense, our energy function behaves akin to a learned dynamic loss [Wu et al., 2018, Bechtle et al., 2019], actively adjusting itself to the prediction network's behavior.

4 EXPERIMENTS

Multi-Label Classification We use the following multi-label classification datasets for testing our model: bibtex [Katakis et al., 2008], delicious [Tsoumakas et al., 2008], eurlexev [Mencia and Fürnkranz, 2008]. The performance metric is F1 score , which is also the score function used for training our models. The max-likelihood loss $\mathcal{L}_{\text{MLE}}$ in this case is given by the multi-label binary cross entropy (MBCE). We use the output of A as a vector of Bernoulli

Method		Dataset			
		BibTex	Delicious	Eurlexev	Bookmark
Slow	SPEN	43.12	26.56	41.75	34.4
	NCE	20.12	16.97	19.50	-
	DVN	42.73	29.71	31.90	37.1
	ALEN	46.4	-	-	38.3
	GSPEN	**48.6**	-	-	**40.7**
Fast	MBCE	42.47	30.12	43.25	33.8
	iALEN	42.8	-	-	37.2
	$\mathcal{L}_{\text{SSVM}}$	44.55	30.34	42.50	37.9
	$\mathcal{L}_{\text{DVN}}$	44.94	28.87	42.35	38.1
	$\mathcal{L}_{\text{CD}}$	45.76	34.50	42.9	**38.5**†
	$\mathcal{L}_{\text{NCE}}$	**46.21**†	**35.12**	**43.49**	**38.5**†

Table 1: Performance of our approach with different objectives (SSVM,CD,NCE) compared to standard multi-label classification (MBCE) and energy based models (SPEN,DVN,NCE). Our implicit gradient trained model significantly outperforms the other approaches. . † denotes statistically significant

variables, and MBCE is then just the sum of logistic losses over the individual components of y.

$$\mathcal{L}_{\text{MLE}} = \sum_{j=1}^{L} -y^j \log((A_\theta(x))^j) - (1-y^j)\log(1-(A_\theta(x))^j)$$

For fair comparison with earlier works on these datasets, we used the energy network design of Belanger and McCallum [2016]. The corresponding energy function E_ϕ is parameterized as:

$$E_\phi = y^T W b(x) + v^T \sigma(My)$$

where the parameters ϕ comprise of $\{W, v, M, b\}$. Network b is defined by a multilayer perceptron. A similar multilayer perceptron from the basis of the inference network A_θ.

We experiment with SPEN [Belanger et al., 2017], DVN [Gygli et al., 2017], and an energy model trained by NCE loss [Gutmann and Hyvärinen, 2010, Ma and Collins, 2018]. As a baseline we also present the results of an MLP trained by standard multi-label binary cross entropy, and ALEN, iALEN [Pan et al., 2020] and GSPEN [Graber and Schwing, 2019]. For our proposed implicit training method, we experiment with different objectives for inner-optimization $\mathcal{L}_{\text{Prim}}$ as described in the section: "Primary Loss Design". Our results are presented in Table 1.

From the experiments, it is clear that our implicit training approach is superior to most current approaches of using energy based models for structured prediction. Our implicit gradient method gives a boost of upto *5 F1 points* depending on the primary loss objective and the dataset. Furthermore, we also note that ($\mathcal{L}_{SSVM}$, SPEN) and ($\mathcal{L}_{DVN}$, DVN) use the same loss and energy function, and the difference in results is attributable to our proposed implicit

Method	RCV		AAPD	
	Mi-F1	Ma-F1	Mi-F1	Ma-F1
SGM	86.9	-	70.2	-
BERT-CE	87.1	66.7	74.1	57.2
OCD	-	-	72.1	58.5
Seq2Seq	87.9	66.0	69.0	54.1
SeqTag	87.7	68.7	73.1	58.5
LACO	88.2	**69.1**	74.7	59.1
$\mathcal{L}_{\text{NCE}}$	**88.5**†	68.9	**75.6**†	**59.8**†

Table 2: Performance of our model on large scale multi-label classification against existing models (SGM, OCD, Seq2Seq, BERT-CE, LACO). Our implicit gradient trained model significantly outperforms or matches other approaches. † denotes statistically significanct scores

training of the inference network. Next, we also note that the only model that outperforms our proposed method is GraphSPEN/GSPEN, which lacks scalability. For example the running-time of GSPEN on Bib (which is our smallest dataset) is more than 6 times our approach. This is due to the need of computationally hard constrained inference in GSPEN and makes it infeasible on the larger datasets that we experiment with in the next section. Finally we see that the noise contrastive objective outperforms the other methods, and so we focus on this objective in our other experiments.

Large Scale Multi-label Modelling . To demonstrate that our approach is more scalable and general, we apply our approach on two large text based datasets RCV1 [Lewis et al., 2004] and AAPD [Yang et al., 2018]. Existing models on these datasets instead rely on standard max likelihood training. The dependence between labels is usually modeled by novel architectures [Zeng et al., 2021], transforming the problem into sequence prediction (SGM) [Yang et al., 2018] or by adding regularization terms to improve representation (LACO) [Zhang et al., 2021]. There are no available energy based baselines on these tasks, partly because of intractability of inference required for energy based structured prediction. We use models from the aforementioned works as baselines, and use a similar architecture to the smaller MLC task for our energy model, except that our feature networks use pretrained BERT models. We also compare to the state of the art LACO model that uses BERT to learn label embeddings [Zhang et al., 2021], the seq2seq approach of Nam et al. [2017] and Tsai and Lee [2020] which is a RNN based auto-regressive decoder. Our results are presented in Table 2. It is clear that using energy based method significantly outperforms BERT based models and edges out ahead of other methods which explicitly focus on modeling label dependence.

Named Entity Recognition . For our experiments we work with the commonly used CoNLL 2003 English dataset

Models	NER	NER+
BILSTM	84.9	89.1
SPEN	85.1	88.6
InfNet	85.2	89.3
InfNet+	85.3	89.7
$\mathcal{L}_{\text{SSVM}}$	85.4	89.6
$\mathcal{L}_{\text{NCE}}$	**85.7**	**90.3**†

Table 3: Test results for NER and NER+ for different energy based models. SSVM and NCE refers to our implicit gradient models. † indicates statistical significance

[Tjong Kim Sang and De Meulder, 2003]. Similar to previous work [Ratinov and Roth, 2009], we consider 17 NER labels, and evaluate the results based on the F1 score. Following Tu and Gimpel [2018], we design the energy network E_ϕ and the inference network A_θ based on Glove based word embeddings [Pennington et al., 2014]. The text embeddings are then provided to bi-LSTMs to form the features $b(x)$ for the energy function. If we denote by $b(x,t)$ the bi-LSTM output at step t, then the energy is :

$$E_\phi(x,y) = \sum_{t=1}^{T}\sum_{j=1}^{L} y_{t,j} U_j^\top b(x,t) + \sum_{t=1}^{T} y_{t-1}^\top W y_t \quad (6)$$

The parameters ϕ compose of the matrix W and the per label parameter U_j, along with the LSTM parameters. Similarly $A_\theta(x)$ can be written as a linear MLP over $b(x)$.

We run our models with two different input feature sets. For the NER version, the input consists of only words and their Glove embeddings. NER+ configuration also provides POS tags and chunk information. As baselines we use SPEN [Belanger et al., 2017], InfNet[Tu and Gimpel, 2018], InfNet+[Tu et al., 2020a] and a cross entropy trained BILSTM baseline. Our results in Table 3 show that implicit models outperform other existing models. Note in particular that our model with SSVM loss is very similar to the InfNet+[Tu et al., 2020a] (with the same losses etc.). The difference between these is a) the final layer in the inference networks F, A are not shared in Infnet+ but are in ours and b) the training procedure is different due to using implicit gradients. If both these models are trained correctly then their final performance should be consistent which seems to be the case. Finally similar to the previous experiments, we see improved performance with contrastive losses.

Citation Field Extraction . We also run our model on the citation-field extraction task [Seymore et al., 1999]. This is an information extraction task where the goal is to segment a citation text into its constituents such as Author, Title, etc. We use the extended Cora citation dataset Seymore et al. [1999] used in Rooshenas et al. [2019]. The citation texts have a max length of 118 tokens, which can be labeled with one of 13 tags.

We explore this task in the indirectly semi-supervised structured prediction of Rooshenas et al. [2019]. In this setting, we have a few labeled points, and are also given rules based rewards for the unlabeled samples. However, the citation reward loss is based on domain knowledge and is noisy. For the task loss, we use token-level accuracy to supplement the reward function. Similarly, the model performance is measured on token-level accuracy. We run this task with 1000 unlabeled and 5, 10, and 50 labeled data points. We compare against GE, RSPEN and SGSPEN [Rooshenas et al., 2019], DVN [Gygli et al., 2017] and our method. Our results are presented in Table 4.

	GE	RSPEN	DVN	SGSPEN	Ours($\mathcal{L}_{NCE}$)
5	54.7	55.0	57.4	53.0	**58.9**
10	57.9	65.0	60.9	62.4	**67.8**
50	68.0	81.5	79.4	82.6	**82.9**

Table 4: Comparing performance of our approach on the semi-supervised setting for the citation-field extraction task. Our implicit gradient trained model significantly outperforms the other approaches.

Ablations The key difference between existing methods like Lee et al. [2021], Tu et al. [2020a] and ours is that in our approach the energy function is updated via the 'true' gradient (Equation 3) while these approaches use alternate optimization and hence use only the explicit gradient term of Equation 6. To demonstrate that these works are less effective due to using biased gradients, we compare our approach to such models in Table 5. For this experiment we focus on the smaller multi-label classification task and experiment with all four objectives discussed earlier. The results show our method to be consistely superior likely because the implicit gradient term provides explicit information to the energy network E_ϕ not only via the output samples of the inference net A_θ, but also via the Hessian of the parameters ϕ.

Time Comparisons In Table 6, we provide the training time and inference time comparison of our method against other methods like SPEN and DVN on multi-label classification datasets. As can be seen the inference time of our proposed method is much better than gradient descent based methods of SPEN and DVN. Moreover , the SOTA GSPEN method of Graber and Schwing [2019] takes more than 13s ($>$ 6 times our approach) for one pass over the bib dataset, highlighting its inefficiency which makes using it infeasible on larger tasks.

Additional Experiments We also conduct experiments with role labeling, POS tagging and image segmentation. These results are available in the supplementary material.

Method	Objective	Dataset		
		BibTex	Delicious	Eurlexev
Non-Implicit	$\mathcal{L}_{\text{SSVM}}$	43.15	28.91	42.10
	$\mathcal{L}_{\text{DVN}}$	42.59	30.17	42.15
	$\mathcal{L}_{\text{CD}}$	43.3	31.09	42.79
	$\mathcal{L}_{\text{NCE}}$	43.2	33.08	42.19
Ours	$\mathcal{L}_{\text{SSVM}}$	44.55	30.34	42.50
	$\mathcal{L}_{\text{DVN}}$	44.94	28.87	42.35
	$\mathcal{L}_{\text{CD}}$	45.76	34.50	42.92
	$\mathcal{L}_{\text{NCE}}$	46.21	35.12	43.49

Table 5: Ablation study of our method using implicit gradients to tune the loss against the SEAL method. Our proposal consistently outperforms as the implicit gradient tunes the energy network towards a loss surface more amenable for the inference net .

	Training Time		Inference Time	
	Bib	Eurlexev	Bib	Eurlexev
SPEN	28.2	134.5	3.8	24.5
DVN	32.1	128.7	3.8	24.6
Ours	27.7	45.6	1.8	12.1

Table 6: Training and inference time (sec/epoch) comparison of our approach against SPEN and DVN. Since the number of parameter update steps for our approach is different per epoch than other models, we have normalized training time/epoch by the number of parameter updates.

5 RELATED WORK

Implicit Gradients Implicit gradients are a powerful technique with a wide range of applications. Recently they have been used for applications like few-shot learning [Rajesvaran et al., 2019, Lee et al., 2019] and building differentiable optimization layers in neural-networks [Amos and Kolter, 2017, Agrawal et al., 2019]. These techniques also arise naturally in other problems related to differentiating through optimizers [Vlastelica et al., 2019], such as general hyper-parameter optimization [Lorraine et al., 2020]. For more detailed review of implicit gradients we refer the readers to Dontchev and Rockafellar [2009], Krantz and Parks [2002]. Implicit gradient methods have been used for energy based learning of MRFs [Tappen et al., 2007, Samuel and Tappen, 2009]. These works have been further extended to use finite-difference methods. While, our work is similar in that it focuses on using implicit gradients for learning energy based models; we focus on structured prediction tasks.

Structured Prediction In recent years energy based models have become prominent in the field [Belanger and McCallum, 2016, Rooshenas et al., 2019, Tu and Gimpel, 2019]. These models essentially relax the output space to a continuous version on which an energy function is learnt for scoring the outputs. Structured prediction energy networks [Belanger et al., 2017, Rooshenas et al., 2019] pair up such energy based models with gradient-based inference for prediction. The training methods for these models have generally relied on generalized version of structural SVM learning [Tsochantaridis et al., 2004], with repeated cost augmented inference being done to adapt the energy models landscape. Due to the difficulty of prediction and instability in training such models Tu and Gimpel [2018] propose an approach called InfNet which directly performs the inference step instead of using gradient descent or other optimization procedures. Our work directly builds upon recent research on energy based structured prediction [Tu et al., 2020a, Lee et al., 2021]. The most important difference between these works and ours is the bi-level optimization formulation and use of implicit gradients. To the best of our knowledge no work in structured prediction literature uses implicit gradient based methods. Secondly, most works either use cost-augmented inference during training [Rooshenas et al., 2019, Belanger and McCallum, 2016] or use the inference network and energy network in an adversarial game [Belanger et al., 2017, Tu and Gimpel, 2018]. The former increases inference time significantly while the latter uses incorrect gradients. ALEN [Pan et al., 2020] propose augmenting the deep energy model of a SPEN with adversarial loss. To handle structural constraints and have direct control over correlations between output variables, Graber and Schwing [2019] incorporate classical inference into SPENs.

An important difference of our method differs from these methods is that we 'meta-learn' the energy function as a trainable objective and can be applied to adjust training of these models as well. Moreover models like GraphSPEN which incorporate constrained inference are not scalable. Our approach side-steps this issue by using an Inference Network [Tu and Gimpel, 2018] approach. Finally ideas from energy based learning have been used in translation [Tu et al., 2020b, Bhattacharyya et al., 2020, Edunov et al., 2017] and text generation [Deng et al., 2020].

Learning Dynamic and Surrogate Losses Surrogate loss learning was formulated as a multi-level optimization by Colson et al. [2007]. Our work uses the insight of Hazan et al. [2010], to interpret learning a structured energy model as a surrogate loss learning problem and uses the bi-level optimization framework to solve the corresponding task. Modern works such as that of Wu et al. [2018], Huang et al. [2019], Bechtle et al. [2019] have attempted to learn dynamic losses for standard classification and regression tasks. Other works such as Sung et al. [2017], Houthooft et al. [2018] have also proposed learning a reward function for optimization. While the goal of these works and ours is similar in that we try to 'learn' an objective loss for increasing a model performance, there are multiple key differences between them. First, these works do not look at the implicit gradient. Instead they rely on 'unrolling' one/few-step

gradient updates in the inner optimization and then back-propagate through those updates. This leads to improper characterization of the model/optimizee parameters induced by the learned loss. Secondly, in the supervised learning based applications the model tries to boost a validation set performance, while in our case we are optimizing the prediction on the training examples via the task loss function available in the structured prediction setting.

Meta Learning Our method has some algorithmic similarities with learning to learn methods [Schmidhuber, 1987]. This is due to the general nature of bi-level objectives which has been adapted for learning hyper-parameters [Franceschi et al., 2018], learning policies for parameter update [Maclaurin et al., 2015, Franceschi et al., 2017, Meier et al., 2018] and meta-learning Rajesvaran et al. [2019]. The key idea in meta-learning is to make the model 'aware' of the 'learning process' [Schmidhuber, 1987, Thrun and Pratt, 2012]. However meta-learning is commonly used for learning model parameters θ that can be easily adapted to new tasks Mendonca et al. [2019], Gupta et al. [2018], multi-task transfer learning [Metz et al., 2019]; while we aim to learn a loss function.

6 CONCLUSION

Summary The primary goal of our work is to learn dynamic losses for model optimization using implicit gradients, in a setting with complex outputs such as in structured prediction. This work uses a bi-level optimization framework for structured prediction that uses a dynamic loss. Then we use implicit gradients to optimize an energy-based model in our proposed framework. We also explore possible designs of these dynamic objectives. Our experiments show our approach outperforms or achieves similar results to existing approaches. Our method tends to be more stable than existing approaches based on inference networks and gradient-based inference.

Limitations and Social Impact Our contributions are mostly restricted to inference network based structured prediction; and our experiments are mostly textual datasets. Structured prediction has also been explored in domains like generative modelling, but our experiments are of little insight into those areas. Moreover, even though our approach trains better than other energy based methods, they are still more sensitive to hyperparameters than standard autoregressive models. We do not foresee any negative societal impact from this work.

Acknowledgements

We thank Yash Chandak and Vihari Piratla for discussions and feedback on the paper.

References

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and Zico Kolter. Differentiable convex optimization layers. *arXiv preprint arXiv:1910.12430*, 2019.

Brandon Amos and Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, 2017.

Sarah Bechtle, Yevgen Chebotar, Artem Molchanov, Ludovic Righetti, Franziska Meier, and Gaurav S Sukhatme. Meta-learning via learned loss. 2019.

David Belanger and Andrew McCallum. Structured prediction energy networks. In *ICML*, 2016.

David Belanger, Bishan Yang, and Andrew McCallum. End-to-end learning for structured prediction energy networks. In *ICML*, 2017.

Sumanta Bhattacharyya, Amirmohammad Rooshenas, Subhajit Naskar, Simeng Sun, Mohit Iyyer, and Andrew McCallum. Energy-based reranking: Improving neural machine translation using energy-based models. *arXiv preprint arXiv:2009.13267*, 2020.

Bruce Christianson. Automatic hessians by reverse accumulation. *IMA Journal of Numerical Analysis*, 1992.

Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153(1), 2007.

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. Residual energy-based models for text generation. *arXiv preprint arXiv:2004.11714*, 2020.

Asen L Dontchev and R Tyrrell Rockafellar. *Implicit functions and solution mappings*, volume 543. Springer, 2009.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. Classical structured prediction losses for sequence to sequence learning. *arXiv preprint arXiv:1711.04956*, 2017.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1165–1173. JMLR. org, 2017.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *ICML*, 2018.

Colin Graber and Alexander Schwing. Graph structured prediction energy networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Information Processing Systems*, 2018.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

Michael Gygli, Mohammad Norouzi, and Anelia Angelova. Deep value networks learn to evaluate and iteratively refine structured outputs. In *International Conference on Machine Learning*, pages 1341–1351. PMLR, 2017.

Tamir Hazan, Joseph Keshet, and David McAllester. Direct loss minimization for structured prediction. *Neurips*, 23, 2010.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, 2016.

Rein Houthooft, Yuhua Chen, Phillip Isola, Bradly C. Stadie, Filip Wolski, Jonathan Ho, and Pieter Abbeel. Evolved policy gradients. In *NeurIPS*, 2018.

Chen Huang, Shuangfei Zhai, Walter Talbott, Miguel Bautista Martin, Shih-Yu Sun, Carlos Guestrin, and Josh Susskind. Addressing the loss-metric mismatch with adaptive loss alignment. In *International Conference on Machine Learning*. PMLR, 2019.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD*, volume 18, page 5. Citeseer, 2008.

Xiangnan Kong, Xiaoxiao Shi, and Philip S Yu. Multi-label collective classification. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 618–629. SIAM, 2011.

Steven George Krantz and Harold R Parks. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002.

Jay-Yoon Lee, Purujit Goyal, Wenlong Zhao, and Zhiyang Xu. Structured energy network for multi-label classification. 2021.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *JMLR*, 5, 2004.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *AISTATS*, 2020.

Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency, 2018.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning*, pages 2113–2122, 2015.

Franziska Meier, Daniel Kappler, and Stefan Schaal. Online learning of a memory for learning rates. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2425–2432. IEEE, 2018.

Eneldo Mencia and Johannes Fürnkranz. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *ECML*. Springer, 2008.

Russell Mendonca, Abhishek Gupta, Rosen Kralev, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Guided meta-policy search. *arXiv preprint arXiv:1904.00956*, 2019.

Luke Metz, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein. Learning unsupervised learning rules. In *ICLR*, 2019.

Jinseok Nam, Eneldo Mencía, Hyunwoo J Kim, and Johannes Fürnkranz. Maximizing subset accuracy with recurrent neural networks in multi-label classification. *NIPS*, 2017.

Pingbo Pan, Ping Liu, Yan Yan, Tianbao Yang, and Yi Yang. Adversarial localized energy network for structured prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5347–5354, 2020.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002.

Jeffrey Pennington, Richard Socher, and Chris Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

Arvind Rajesvaran, Chelea Finn, Sham Kakade, and Sergey Levin. Meta-learning with implicit gradients. 2019.

Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 2009.

Amirmohammad Rooshenas, Dongxu Zhang, Gopal Sharma, and Andrew McCallum. Search-guided, lightly-supervised training of structured prediction energy networks. In *NIPS 32*. 2019.

Kegan GG Samuel and Marshall F Tappen. Learning optimized map estimates in continuously-valued mrf models. In *CVPR*. IEEE, 2009.

Juergen Schmidhuber. Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook. Institut für Informatik, Technische Universität München, 1987.

Kristie Seymore, Andrew McCallum, Roni Rosenfeld, et al. Learning hidden markov model structure for information extraction. In *AAAI-99 workshop on machine learning for information extraction*, pages 37–42, 1999.

Lili Song and Luís Nunes Vicente. Modeling hessian-vector products in nonlinear optimization: new hessian-free methods. *IMA Journal of Numerical Analysis*, 2022.

Flood Sung, Li Zhang, Tao Xiang, Timothy Hospedales, and Yongxin Yang. Learning to learn: Meta-critic networks for sample efficient learning. *arXiv preprint arXiv:1706.09529*, 2017.

Marshall F Tappen, Ce Liu, Edward H Adelson, and William T Freeman. Learning gaussian conditional random fields for low-level vision. In *CVPR*. IEEE, 2007.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *NAACL*, 2003.

Che-Ping Tsai and Hung-Yi Lee. Order-free learning alleviating exposure bias in multi-label classification. In *AAAI*, 2020.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of 21st ICML*, 2004.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *MMD'08*, volume 21, 2008.

Lifu Tu and Kevin Gimpel. Learning approximate inference networks for structured prediction. In *ICLR*, 2018.

Lifu Tu and Kevin Gimpel. Benchmarking approximate inference methods for neural structured prediction. *arXiv:1904.01138*, 2019.

Lifu Tu, Richard Yuanzhe Pang, and Kevin Gimpel. Improving joint training of inference networks and structured prediction energy networks, 2020a.

Lifu Tu, Richard Yuanzhe Pang, Sam Wiseman, and Kevin Gimpel. ENGINE: Energy-based inference networks for non-autoregressive machine translation. In *ACL*, 2020b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017.

Francisco Vázquez, José-Jesús Fernández, and Ester M Garzón. A new approach for sparse matrix vector product on nvidia gpus. *Concurrency and Computation: Practice and Experience*, 23(8), 2011.

Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. Differentiation of blackbox combinatorial solvers. *arXiv:1912.02175*, 2019.

Sebastian J Vollmer, Konstantinos C Zygalakis, and Yee Whye Teh. Exploration of the (non-) asymptotic bias and variance of stochastic gradient langevin dynamics. *JMLR*, 17(1), 2016.

Lijun Wu, Fei Tian, Yingce Xia, Yang Fan, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. Learning to teach with dynamic loss functions. *arXiv:1810.12081*, 2018.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. Sgm: sequence generation model for multi-label classification. *arXiv preprint arXiv:1806.04822*, 2018.

Zhizhong Zeng, Yufen Liu, Wenpeng Gao, Baihong Li, Ting Zhang, Xinguo Yu, and Zongkai Yang. Modeling label correlations implicitly through latent label encodings for multi-label text classification. 2021.

Ximing Zhang, Qian-Wen Zhang, Zhao Yan, Ruifang Liu, and Yunbo Cao. Enhancing label correlation feedback in multi-label classification via multi-task learning, 2021.

Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods, 2018.