Learning Randomly Perturbed Structured Predictors for Direct Loss Minimization

Hedda Cohen Indelman [1] **Tamir Hazan** [1]

Abstract

Direct loss minimization is a popular approach for learning predictors over structured label spaces. This approach is computationally appealing as it replaces integration with optimization and allows to propagate gradients in a deep net using loss-perturbed prediction. Recently, this technique was extended to generative models, by introducing a randomized predictor that samples a structure from a randomly perturbed score function. In this work, we interpolate between these techniques by learning the variance of randomized structured predictors as well as their mean, in order to balance between the learned score function and the randomized noise. We demonstrate empirically the effectiveness of learning this balance in structured discrete spaces.

1. Introduction

Learning and inference in high-dimensional structured models drives much of the research in machine learning applications, from computer vision, natural language processing, to computational chemistry. Examples include scene understanding (Kendall et al., 2017) machine translation (Wiseman and Rush, 2016) and molecular synthesis (Jin et al., 2020). The learning process optimizes a score for each of the exponentially many structures in order to best fit the mapping between input and output in the training data. While it is often computationally infeasible to evaluate the loss of all exponentially many structures simultaneously through sampling, it is often feasible to predict the highest scoring structure efficiently in many structured settings.

Direct loss minimization is an appealing approach in discriminative learning that allows to learn a structured model by predicting the highest scoring structure (Hazan et al., 2010; Keshet et al., 2011; Song et al., 2016). It allows to improve the loss of the structured predictor by considering the gradients of two predicted structures: over the original loss function and over a perturbed loss function. This approach implicitly uses the data distribution to smooth the loss function between a training structure and a predicted structure, thus propagating gradients through the maximal argument of the predicted structure. Unfortunately, our access to the data distribution is limited and we cannot reliably represent the intricate relation between a training instance and its exponentially many structures. Recently, this framework was extended to generative learning, where a random perturbation that follows the Gumbel distribution law allows to sample from all possible structures (Lorberbom et al., 2018). However, one cannot apply this generative learning approach effectively to discriminative learning, since the random noise that is added in the generation process interferes in predicting the best scoring structure.

In this work we combine these two approaches: we explicitly add random perturbation to each of the structures, in order to reliably represent the intricate relation between the a training instance and its exponentially many structures. To balance between the learned score function and the added random perturbation, we treat the score function as the mean of the random perturbation, and learn its variance. This way we are able to control the ratio between the signal (the score) and the noise (the random perturbation) in discriminative learning.

In summary, we make the following contributions:

1. We show that the uniqueness assumption of the predicted structure is a key element in the gradient step of direct loss minimization, thus mathematically defining its general position assumption.
2. We prove that random perturbation ensures unique maximizers with probability one.
3. We identify that random perturbation might also serve as noise that masks the score function signal. Hence, we introduce a method for learning both the mean and the variance of randomized predictors in the high-dimensional structured label setting.
4. We show empirically the benefit of our approach in two structured prediction problems.

[1]Technion. Correspondence to: Hedda Cohen Indelman <cohen.hedda@campus.technion.ac.il>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

2. Related Work

Effective structured learning and inference over discrete, combinatorial models is challenging and has been addressed by different approaches.

Direct loss minimization is an effective approach in discriminative learning that was devised to optimize non-convex and non-smooth loss functions for linear structured predictors (Hazan et al., 2010). Later it was extended to non-linear models, including hidden Markov models and deep learners (Keshet et al., 2011; Song et al., 2016). Our work extends direct loss minimization by adding random noise to its structured predictor and learning its variance. Recently, the idea of optimization that replaces sampling was extended to generative learning and reinforcement learning (Lorberbom et al., 2018; 2019). Similar to our work, these works also add random Gumbel perturbation and learn the mean of their structured predictor. In contrast, our work also learns the variance of the predictor, and our experimental validation shows it contributes to the performance of the predictor. Also, our theoretical contribution sets the framework to handle any structured predictor. Closely related is a method of differentiating through marginal inference (Domke, 2010), which shows that the gradient of the loss with respect to the parameters can be computed based on inference over the original parameters , and one over the parameters pertubed in the direction of the loss derivative w.r.t. to the marginals.

Another line of work considers continuous relaxations of the discrete structures. Paulus et al. (2020) have suggested a unified framework for constructing structured relaxations of combinatorial distributions, and have demonstrated it as a generalization of the Gumbel-Softmax trick. Their method builds upon differentiating through a convex program and induces solutions found in the interior of the polytope rather than on its faces, as a function of temperature-controlled approximation. An efficient extension for sorting and ranking differential operators has been suggested lately (Blondel et al., 2020). SparseMAP (Niculae et al., 2018) is a sparse structured inference framework which offers a continuous relaxation. It finds sparse MAP solutions on the faces of the marginal polytope. Recently, Berthet et al. (2020) suggested stochastic smoothing to allow differentiation through perturbed maximizers. In contrast, we do not use convex smoothing techniques of the structured label for differentiation.

Blackbox optimization (Pogancic et al., 2020) is a new scheme to differentiate through argmax, which allows backward pass through blackbox implementations of combinatorial solvers with linear objective functions.

Our work considers two popular structured prediction problems: bipartite matching and k-nearest neighbors. Learning matchings in bipartite graphs has been extensively researched. When the bipartite graph is balanced, a matching can be represented by a permutation, which is an extreme point of the Birkhoff polytope, i.e., the set of all doubly stochastic matrices. Many works have built upon Sinkhorn normalization, an algorithm that maps a square matrix to a doubly-stochastic matrix. The Sinkhorn normalization has been incorporated in end-to-end learning algorithms in order to obtain relaxed gradients for learning to rank (Adams and Zemel, 2011), bipartite matching (Mena et al., 2018), visual permutation learning (Santa Cruz et al., 2019), and latent permutation inference (Linderman et al., 2018). This continuous relaxation is inspired by the Gumbel-Softmax trick (Jang et al., 2016; Maddison et al., 2017). Andriyash et al. Andriyash et al. (2018) have later showed that the Gumbel-Softmax estimator is biased and proposed a method to reduce its bias. We also consider the problem of stochastic maximization over the set of possible latent permutations. However, we do not relax the use of bipartite matchings. Instead, we directly optimize the bipartite matching predictor and propagate gradients using the direct optimization approach.

Our work also considers learning k-nearest neighbors, i.e., learning an embedding of points that encourages the k closest points to the test point to have the correct label. The body of work on sorting and specifically top-k operators in an end-to-end learning framework is extensive. Grover et al. (2019) have suggested a continuous relaxation of the output of the sorting operator from permutation matrices to the set of unimodal row-stochastic matrices, where every row sums to one and has a distinct maximal argument. Plötz and Roth (2018) developed a continuous deterministic relaxation that maintains differentiability with respect to pairwise distances, but retains the original k-nearest neighbors as the limit of a temperature parameter approaching zero. Other approaches are based on top-k subset sampling (Xie and Ermon, 2019; Kool et al., 2019). Berrada et al. (2018) have introduce a family of smoothed, temperature controlled loss functions that are suited to top-k optimization. In contrast, our work does not relax the objective but rather directly optimize the top-k neighbors. Xie et al. (2020) have proposed a smoothed approximation to the top-k operator as the solution of an Entropic Optimal Transport problem.

3. Background

Learning to predict structured labels $y \in Y$ of data instances $x \in X$ covers a wide range of problems. The structure is incorporated into the label $y = (y_1, ..., y_n)$ which may refer to matchings, permutations, sequences, or other high-dimensional objects. For any data instance x, its different structures are scored by a parametrized function $\mu_w(x, y)$. Discriminative learning aims to find a mapping from training data $S = \{(x_1, y_1), ..., (x_m, y_m)\}$ to parameters w for

which $\mu_w(x,y)$ assign high scores to structures y that describe well the data instance x. The parameters w are fitted to minimize the loss $\ell(\cdot,\cdot)$ of the instance-label pairs $(x,y) \in S$ between the label y and the highest scoring structure of $\mu_w(x,y)$. While gradient methods are the most popular methods to learn the parameters w, they are notoriously inefficient for learning discrete predictions. When considering discrete labels, the maximal argument of $\mu_w(x,y)$ is a piecewise constant function of w, and its gradient with respect to w is zero for almost any w. Consequently, various smoothing techniques were proposed to propagate gradients while learning to fit discrete structures.

Direct loss minimization approach aims at minimizing the expected loss $\min_w \mathbb{E}_{(x,y)\sim\mathcal{D}}\ell(y^*_w, y)$ that incurs when the training label y is different than the predicted label (Hazan et al., 2010; Keshet et al., 2011; Song et al., 2016)

$$y^*_w \triangleq \arg\max_{\hat{y}} \mu_w(x,\hat{y}) \tag{1}$$

Direct loss minimization relies on a loss-perturbed prediction

$$y^*_w(\epsilon) \triangleq \arg\max_{\hat{y}} \{\mu_w(x,\hat{y}) + \epsilon\ell(y,\hat{y})\}. \tag{2}$$

It introduces an optimization-based gradient step for the expected loss, namely $\nabla\mathbb{E}_{(x,y)\sim\mathcal{D}}\ell(y^*_w, y) =$

$$\lim_{\epsilon\to 0}\frac{1}{\epsilon}\Big(\mathbb{E}_{(x,y)\sim\mathcal{D}}[\nabla_w\mu_w(x,y^*_w(\epsilon)) - \nabla_w\mu_w(x,y^*_w)]\Big). \tag{3}$$

Unfortunately, the above gradient step does not hold for any w, cf. (Hazan et al., 2010) Section 3.1. For example, when $w = 0$ the gradient estimator in Equation (3) may be zero for any $(x,y) \sim \mathcal{D}$ regardless of the value of $\nabla\mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(y^*_w, y)]$. In Section 4.1 we define the mathematical condition for which Equation (3) represents the gradient.

Recently, the direct loss minimization technique was applied to generative learning. In this setting, a random perturbation $\gamma(y)$ is added to each configuration, (Lorberbom et al., 2018). The technique allows to randomly generate structures y for any given x from a generative distribution $q(y|x) \propto e^{\mu_w(x,y)}$. The generative learning approach relies on the connection between $q(y|x)$ and the Gumbel-max trick, namely $\mathbb{P}_{\gamma\sim g}[y^*_{w,\gamma} = y] \propto e^{\mu_w(x,y)}$, when $y^*_{w,\gamma} = \arg\max_{\hat{y}}\{\mu_w(x,\hat{y}) + \gamma(\hat{y})\}$ and $\gamma(y)$ are i.i.d. random variables that follow the zero mean Gumbel distribution law, which we denote by $\mathcal{G}$. The corresponding gradient step, in discriminative learning setting, takes the form: $\nabla\mathbb{E}_{\gamma\sim\mathcal{G}}[\ell(y^*_{w,\gamma}, y)] =$

$$\lim_{\epsilon\to 0}\frac{1}{\epsilon}\Big(\mathbb{E}_{\gamma\sim\mathcal{G}}[\nabla\mu_w(x,y^*_{w,\gamma}(\epsilon)) - \nabla\mu_w(x,y^*_{w,\gamma})]\Big). \tag{4}$$

Here, $y^*_{w,\gamma}(\epsilon) = \arg\max_{\hat{y}}\{\mu_w(x,\hat{y}) + \gamma(\hat{y}) + \epsilon\ell(y,\hat{y})\}$. The advantage of using this framework in this setting is that it effortlessly elevates the mathematical difficulties in defining the gradient of the expected loss that exists in the direct loss minimization framework. Unfortunately, the random noise $\gamma(y)$ that is injected to the optimization may mask the signal $\mu_w(x,y)$ and thus get sub-optimal results in discriminative learning. To enjoy the best of both worlds, we propose to learn the proper amount of randomness to add to the discriminative learner.

In this work we focus on learning discrete structured labels $y = (y_1, ..., y_n)$. A general score function $\mu_w(x,y)$ cannot be computed efficiently for discrete structured labels $y = (y_1, ..., y_n)$ since the number of possible labels is exponential in n and a general score function $\mu_w(x,y)$ may assign a different value for each structure. Typically, such score functions are decomposed to localized score functions over small subsets $\alpha \subset \{1, ..., n\}$ of variables where $y_\alpha = (y_i)_{i\in\alpha}$. The score function takes the form: $\mu_w(x,y) = \sum_{\alpha\in A}\mu_{w,\alpha}(x, y_\alpha)$. The correspondence between the exponential family of distributions $e^{\mu_w(x,y)}$ and the Gumbel-max trick requires an independent random variable $\gamma(y)$ for each of the exponentially many structures $y = (y_1, ., , , .y_n)$. However, since we are focusing on discriminative learning we are not limited by the Gumbel-max trick. Instead, we can use fewer random variables in order to learn the minimal amount of randomness to add. We limit our predictors to low-dimensional independent random variables $\gamma(y) = \sum_{i=1}^n \gamma_i(y_i)$, where $\gamma_i(y_i)$ are independent random variables for each index $i = 1, ..., n$ and each y_i. In this setting, the number of random variables we are using is linear in n, compared to exponential many random variables in the Gubeml-max setting.

4. Learning Structured Predictors

In the following we present our main technical concept that derives the gradient of an expected loss using two structured predictions. In Section 4.1 we prove the gradient step of an expected loss in the direct loss minimization framework. We also deduce that it holds whenever $y^*_w(\epsilon)$ is unique. Subsequently, in Section 4.2, we show that low dimensional random perturbations $\gamma_i(y_i)$ are able to implicitly enforce uniqueness of the maximizing structure with probability one. In Section 4.3, we present our approach that learns the variance of the random perturbation, to ensure that the random noise $\gamma_i(y_i)$ does not mask the signal $\mu_w(x,y)$.

4.1. Direct Loss Minimization

We rely on the expected max-value that is perturbed by the loss function. This is the "prediction generating function" in Lorberbom et al. (2018). In the direct loss minimization setting, as defined in Equation (3), this function takes the

form:

$$G(w,\epsilon) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\max_{\hat{y}\in Y}\big\{\mu_w(x,\hat{y}) + \epsilon\ell(y,\hat{y})\big\}\Big] \quad (5)$$

The proof technique relies on the existence of the Hessian of $G(w,\epsilon)$ and the main challenge is to show that $G(w,\epsilon)$ is differentiable, i.e., there exists a vector $\nabla\mu_w(x,y^*(\epsilon))$ such that for any direction u, its corresponding directional derivative $\lim_{h\to 0}\frac{G(w+hu,\epsilon)-G(w,\epsilon)}{h}$ equals $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\nabla_w\mu_w(x,y^*(\epsilon))^\top u]$. The proof builds a sequence of functions $\{g_n(u)\}_{n=1}^\infty$ that satisfies

$$\lim_{h\to 0}\frac{G(w+hu,\epsilon)-G(w,\epsilon)}{h} = \lim_{n\to\infty}\mathbb{E}_{(x,y)\sim\mathcal{D}}[g_n(u)] \quad (6)$$

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[\lim_{n\to\infty} g_n(u)] = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\nabla_w\mu_w(x,y^*(\epsilon))^\top u]. \quad (7)$$

The functions $g_n(u)$ correspond to the loss perturbed prediction $y^*_w(\epsilon)$ through the quantity $\mu_{w+\frac{1}{n}u}(x,\hat{y}) + \epsilon\ell(y,\hat{y})$. The key idea we are exploiting is that there exists n_0 such that for any $n \geq n_0$ the maximal argument $y^*_{w+\frac{1}{n}u}(\epsilon)$ does not change.

Lemma 1. *Assume $\mu_w(x,y)$ are continuous functions of w and that their loss-perturbed maximal argument $y^*_{w+\frac{1}{n}u}(\epsilon)$, which is defined in Equation (2), is unique for any u and n. Then there exists n_0 such that for $n \geq n_0$ there holds $y^*_{w+\frac{1}{n}u}(\epsilon) = y^*_w(\epsilon)$.*

Proof. Let $f_n(y) = \mu_{w+\frac{1}{n}u}(x,y) + \epsilon\ell(y,\hat{y})$ so that $y^*_{w+\frac{1}{n}u}(\epsilon) = \arg\max_y f_n(y)$. Also, let $f_\infty(y) = \mu_w(x,y)+\epsilon\ell(y,\hat{y})$ so that $y^*_w(\epsilon) = \arg\max_y f_\infty(y)$. Since f_n is a continuous function of then $\max_y f_n(y)$ is also a continuous function and $\lim_{n\to\infty}\max_y f_n(y) = \max_y f_\infty(y)$. Since $\max_y f_n(y) = f_n(y^*_{w+\frac{1}{n}u}(\epsilon))$ is arbitrarily close to $\max_y f_\infty(y) = f_\infty(y^*_w(\epsilon))$, and $y^*_w(\epsilon), y^*_{w+\frac{1}{n}u}(\epsilon)$ are unique then for any $n \geq n_0$ these two arguments must be the same, otherwise there is a $\delta > 0$ for which $|f_\infty(y^*_w(\epsilon)) - f_n(y^*_{w+\frac{1}{n}u}(\epsilon))| \geq \delta$. □

This lemma relies on the discrete nature of the label space, ensuring that the optimal label does not change in the vicinity of $y^*_w(\epsilon)$. This phenomena distinguishes the discrete label setting from the continuous relaxations of the label space (Domke, 2010; Berthet et al., 2020; Paulus et al., 2020). These relaxations of the label space utilize their continuities to differentiate through the label. In direct loss minimization, one works directly with the discrete label space which allows to control the maximal argument in infinitesimal interval.

Theorem 1. *Assume $\mu_w(x,y)$ is a smooth function of w and that $E_{(x,y)\sim\mathcal{D}}\|\nabla_w\mu_w(x,y)\| \leq \infty$. If the conditions of Lemma 1 hold then the prediction generating function $G(w,\epsilon)$, as defined in Equation (5), is differentiable and*

$$\frac{\partial G(w,\epsilon)}{\partial\epsilon} = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(y,y^*_w)]. \quad (8)$$

$$\frac{\partial G(w,\epsilon)}{\partial w} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\nabla\mu_w(x,y^*(\epsilon))\Big]. \quad (9)$$

Proof. Let $f_n(y) = \mu_{w+\frac{1}{n}u}(x,y) + \epsilon\ell(y,\hat{y})$ as in Lemma 1 and let

$$g_n(u) \triangleq \frac{\max_{\hat{y}\in Y} f_n(\hat{y}) - \max_{\hat{y}\in Y} f_\infty(\hat{y})}{1/n} \quad (10)$$

We apply the dominated convergence theorem on $g_n(u)$, so that $\lim_{n\to\infty}\mathbb{E}_{(x,y)\sim\mathcal{D}}[g_n(u)] = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\lim_{n\to\infty} g_n(u)]$ in order to prove Equations (6,7). We note that we may apply the dominated convergence theorem, since the conditions $\mathbb{E}_{(x,y)\sim D}\|\nabla_w\mu_w(x,y)\| \leq \infty$ imply that the expected value of g_n is finite (We recall that f_n is a measurable function, and note that since $\hat{y} \in Y$ is an element from a discrete set Y, then g_n is also a measurable function.).

From Lemma 1, the terms $\ell(y,y^*(\epsilon))$ are identical in both $\max_{\hat{y}\in Y} f_n(\hat{y})$ and $\max_{\hat{y}\in Y} f_\infty(\hat{y})$. Therefore, they cancel out when computing the difference $\max_{\hat{y}\in Y} f_n(\hat{y}) - \max_{\hat{y}\in Y} f_\infty(\hat{y})$. Then, for $n \geq n_0$:

$$\max_{\hat{y}\in Y} f_n(\hat{y}) - \max_{\hat{y}\in Y} f_\infty(\hat{y}) = \mu_{w+\frac{1}{n}u}(x,y^*(\epsilon)) - \mu_w(x,y^*(\epsilon))$$

and Equation (10) becomes:

$$g_n(u) = \frac{\mu_{w+\frac{1}{n}u}(x,y^*(\epsilon)) - \mu_w(x,y^*(\epsilon))}{1/n}. \quad (11)$$

Since $\mu_w(x,y^*(\epsilon))$ is smooth, then $\lim_{n\to\infty} g_n(u)$ is composed of the derivatives of $\mu_w(x,y^*(\epsilon))$ in direction u, namely, $\lim_{n\to\infty} g_n(u) = \nabla_w\mu_w(x,y^*(\epsilon))^\top u$. □

In the above theorem we assume that $\mu_w(x,y)$ is smooth, namely it is infinitely differentiable. It suffices to assume that $\mu_w(x,y)$ is twice differentiable, to ensure that $G(w,\epsilon)$ is twice differentiable and hence its Hessian exists.

Corollary 1. *Under the conditions of Theorem 1, $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(y,y^*_w)]$ is differentiable and its derivative is defined in Equation (3).*

Proof. Since Theorem 1 holds for every direction u:

$$\frac{\partial G(w,\epsilon)}{\partial w} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\nabla_w\mu_w(x,y^*(\epsilon))\Big].$$

Adding a derivative with respect to ϵ we get:

$$\frac{\partial}{\partial\epsilon}\frac{\partial G(w,0)}{\partial w} = \lim_{\epsilon\to 0}\frac{1}{\epsilon}\mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\nabla_w\mu_w(x,y^*(\epsilon)) - \nabla_w\mu_w(x,y^*)\Big]$$

The proof follows by showing that the gradient computation is apparent in the Hessian, namely Equation (3) is attained by the identity $\partial_w \partial_\epsilon G(w,0) = \partial_\epsilon \partial_w G(w,0)$. Now we turn to show that $\partial_w \partial_\epsilon G(w,0) = \nabla_w \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(y, y^*_w)]$. Since ϵ is a real valued number rather than a vector, we do not need to consider the directional derivative, which greatly simplifies the mathematical derivations. We define $f_n(\hat{y}) \triangleq \mu_w(x,\hat{y}) + \frac{1}{n}\ell(y,\hat{y})$ and follow the same derivation as above to show that $\partial_\epsilon G(w,0) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(y, y^*_w)]$. Therefore $\partial_w \partial_\epsilon G(w,0) = \nabla_w \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(y, y^*_w)]$. □

We note the strong conditions that require the theorem to hold: the loss-perturbed maximal argument $y^*_{w+\frac{1}{n}u}(\epsilon)$, which is defined in Equation (2), is unique for any u and n. Unfortunately, this condition does not hold in some cases, e.g., when $w = 0$. Next we show that with added random perturbation we can ensure this holds with probability one.

4.2. Randomly Perturbing Structured Predictors

We turn to show that randomly perturbing the structured signal $\mu_w(x,y) = \sum_{\alpha\in\mathcal{A}} \mu_\alpha(x, y_\alpha)$ with smooth random noise $\gamma_i(y_i)$ allows us to implicitly enforce the uniqueness condition. To account for the structured signal and the low-dimensional random perturbation we define the set $y^*_{w,\gamma}(\epsilon) =$

$$\arg\max_{\hat{y}} \Big\{ \sum_{\alpha\in\mathcal{A}} \mu_{w,\alpha}(x, \hat{y}_\alpha) + \sum_{i=1}^{n} \gamma_i(\hat{y}_i) + \epsilon\ell(y,\hat{y}) \Big\}. \quad (12)$$

To reason about the set of maximal structures of $y^*_{w,\gamma}(\epsilon)$, we introduce the set of random perturbation $\Gamma_\epsilon(y')$ which consists of all random values $\gamma_i(y_i)$ for which y' is their maximal structure: $\Gamma_\epsilon(y') =$

$$\left\{ \gamma \;:\; \begin{array}{l} \sum_{\alpha\in\mathcal{A}} \mu_{w,\alpha}(x, \hat{y}'_\alpha) + \sum_{i=1}^{n} \gamma_i(\hat{y}'_i) + \epsilon\ell(y,\hat{y}') \\ \geq \\ \forall \hat{y} \sum_{\alpha\in\mathcal{A}} \mu_{w,\alpha}(x, \hat{y}_\alpha) + \sum_{i=1}^{n} \gamma_i(\hat{y}_i) + \epsilon\ell(y,\hat{y}) \end{array} \right\} \quad (13)$$

Whenever the set in Equation (12) consists of more than a single structure, say y' and y'', their corresponding sets $\Gamma_\epsilon(y')$ and $\Gamma_\epsilon(y'')$ intersect. We now prove that this happens with zero probability whenever $\gamma_i(y_i)$ are i.i.d. and with a smooth probability density function.

Theorem 2. *Let $\gamma_i(y_i)$ be i.i.d. random variables with a smooth probability density function. Then the set of maximal arguments in Equation (12) consists of a single structure with probability one.*

Proof. Consider there is an event (a set) of γ for which the set of maximal arguments consists of more than one structure, e.g., y' and y'' and denote it by Ω. Clearly, $\Omega \subset \Gamma_\epsilon(y') \cap \Gamma_\epsilon(y'')$. Let β be the set of indexes for which $y'_i \neq y''_i$. Since for any $\gamma \in \Omega$ it holds that $\sum_{\alpha\in\mathcal{A}} \mu_{w,\alpha}(x,\hat{y}'_\alpha) + \sum_{i=1}^{n} \gamma_i(\hat{y}'_i) + \epsilon\ell(y,\hat{y}') = \sum_{\alpha\in\mathcal{A}} \mu_{w,\alpha}(x,\hat{y}''_\alpha) + \sum_{i=1}^{n} \gamma_i(\hat{y}''_i) + \epsilon\ell(y,\hat{y}'')$, then $\Omega \subset \{\gamma : \sum_{i\in\beta} \gamma_i(\hat{y}'_i) - \gamma_i(\hat{y}''_i) = c\}$ for the constant $c = \mu_{w,\alpha}(x,\hat{y}''_\alpha) - \mu_{w,\alpha}(x,\hat{y}'_\alpha) + \epsilon\ell(y,\hat{y}'') - \epsilon\ell(y,\hat{y}')$. Since $\gamma_i(\hat{y}'_i) - \gamma_i(\hat{y}''_i)$ are independent random variables with smooth probability density function, then their sum also has a smooth probability density function. Consequently the probability that $\sum_{i\in\beta} \gamma_i(\hat{y}'_i) - \gamma_i(\hat{y}''_i) = c$ is zero, and thus $\mathbb{P}_\gamma[\Omega] = 0$. □

We note that the uniqueness of the maximal structure of $y^*_{w,\gamma}$ can be proved by Theorem 2 as well, in which case the constant $c = \mu_{w,\alpha}(x,\hat{y}''_\alpha) - \mu_{w,\alpha}(x,\hat{y}'_\alpha)$. It follows from the above theorem that adding random perturbations solves the uniqueness problem in direct loss gradient rule. Unfortunately, as we show in our experimental evaluation, the random perturbation that smooths the objective can also serve as noise that masks the signal $\mu_w(x,y)$. To address this caveat, we propose to learn the magnitude, i.e., the variance, of this noise explicitly.

4.3. Learning The Variance Of Randomly Perturbed Structured Predictors

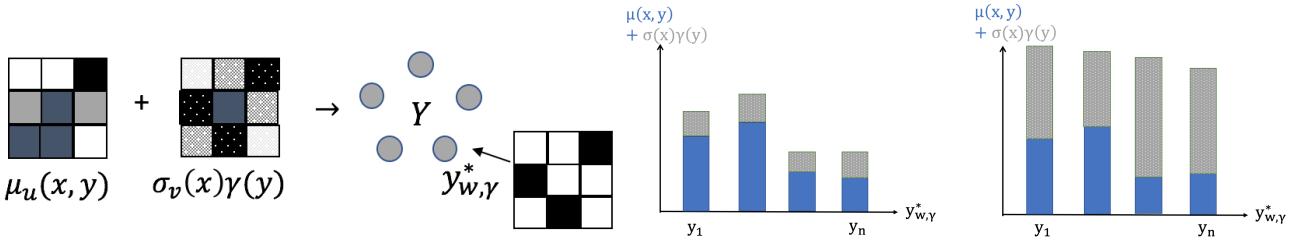
We propose to learn the magnitude of the random perturbation $\gamma_i(y_i)$. In our setting it translates to the prediction $y^*_{w,\gamma} =$

$$\arg\max_{\hat{y}} \Big\{ \sum_{\alpha\in\mathcal{A}} \mu_{u,\alpha}(x, \hat{y}_\alpha) + \sum_{i=1}^{n} \sigma_v(x)\gamma_i(\hat{y}_i) \Big\} \quad (14)$$

$w = (u, v)$ are the learned parameters. In this case we treat $\sum_{\alpha\in\mathcal{A}} \mu_{u,\alpha}(x,\hat{y}_\alpha) + \sum_{i=1}^{n} \sigma_v(x)\gamma_i(\hat{y}_i)$ as a random variable whose mean is learned using $\mu_{u,\alpha}(x,\hat{y}_\alpha)$ and its variance is learned using $\sigma_v(x)$. As such, we consider a strictly positive $\sigma_v(x)$ both theoretically and practically. We are learning the same variance $\sigma_v(x)$ for all random assignments $\gamma_i(y_i)$. We do so to learn to balance the overall noise $\sum_{i=1}^{n} \gamma_i(y_i)$ with the signal $\sum_\alpha \mu_{u,\alpha}(x,\hat{y}_\alpha)$. The learned variance $\sigma_v(x)$ allows us to interpolate between the original direct loss setting, where $\sigma_v(x) = 0$, to the generative learning setting, where $\sigma_v(x) = 1$.

Corollary 2. *Assume $\mu_u(x,y), \sigma_v(x)$ are smooth functions of $w = (u, v)$. Let $\gamma_i(y_i)$ be i.i.d. random variables with a smooth probability density function. Let $G(w,\epsilon) =$*

$$\mathbb{E}_{\gamma\sim\mathcal{G}}\Big[\max_{\hat{y}\in Y} \Big\{ \sum_{\alpha\in\mathcal{A}} \mu_{u,\alpha}(x,\hat{y}_\alpha) + \sum_{i=1}^{n} \sigma_v(x)\gamma_i(\hat{y}_i) + \epsilon\ell(y,\hat{y}) \Big\} \Big]. \quad (15)$$



(a) A randomly perturbed structured prediction illustration

(b) The max predictor probability distribution for a low $\sigma(x)$.

(c) The max predictor probability distribution for a high $\sigma(x)$.

Figure 1. The randomized predictor $y^*_{w,\gamma}$ is the structure that maximizes the randomly perturbed scoring function among all possible structures in Y (Figure 1a). As $\sigma(x)$ decreases, the expected max predictor approaches the expected value of a categorical random variable (Figure 1b). And vice versa, as $\sigma(x)$ increases, the expected max predictor converges to a uniform distribution over the discrete structures (Figure 1c).

Then $G(w,\epsilon)$ *is a smooth function and* $\frac{\partial}{\partial u}\mathbb{E}_\gamma[\ell(y,y^*_{w,\gamma})] =$

$$\lim_{\epsilon\to 0}\frac{1}{\epsilon}\mathbb{E}_\gamma\Big[\sum_{\alpha\in\mathcal{A}}\left(\nabla\mu_{u,\alpha}(x,y^*_\alpha(\epsilon))-\nabla\mu_{u,\alpha}(x,y^*_\alpha)\right)\Big] \quad (16)$$

and $\frac{\partial}{\partial v}\mathbb{E}_\gamma[\ell(y,y^*_{w,\gamma})] =$

$$\lim_{\epsilon\to 0}\frac{1}{\epsilon}\mathbb{E}_\gamma\Big[\sum_{i=1}^{n}\nabla\sigma_v(x)\Big(\gamma_i(y^*_i(\epsilon))-\gamma_i(y^*_i)\Big)\Big]. \quad (17)$$

We prove Corollary 2 in the supplementary material.

The random perturbation induces a probability distribution over structures y. As σ increases, the expected max predictor tends to a uniform distribution over the discrete structures. Similarly, as σ decreases, the expected max predictor approaches a deterministic decision over the discrete structures. This idea is illustrated in Figure 1.

Interestingly, whenever the random variables $\gamma(y)$ follow the zero mean Gumbel distribution law, the random variable $\mu_u(x,\hat{y})+\sigma_v(x)\gamma(\hat{y})$ follows the Gumbel distribution law with mean $\mu_u(x,y)$ and variance $\sigma^2\pi^2/6$. In this case, the variance turns to be the temperature of the corresponding Gibbs distribution: $\mathbb{P}_{\gamma\sim\mathcal{G}}[\arg\max_{\hat{y}}\{\mu_u(x,\hat{y})+\sigma_v(x)\gamma(\hat{y})\}=y]\propto e^{\mu_u(x,y)/\sigma_v(x)}$, see proof in the supplementary material. Our framework thus also allows to learn the temperature of the Gumbel-max trick instead of tuning it as a hyper-parameter.

5. Experimental Validation

In the following we validate the advantage of our approach (referred to as 'Direct Stochastic Learning') in two popular structured prediction problems: bipartite matching and k-nearest neighbors. We compare to direct loss minimization (Hazan et al., 2010), which can be interpreted as setting the noise variance to zero (referred to as Direct $\bar{\sigma}=0$), as well as to Lorberbom et al. (2018), in which the noise variance is set to one (referred to as Direct $\bar{\sigma}=1$). Additionally, we compare to state-of-the-art in neural sorting (Grover et al., 2019; Xie and Ermon, 2019) and bipartite matching (Mena et al., 2018). Further architectural and training details are described in the supplementary material.

In all direct loss based experiments we set a negative ϵ. When $\epsilon>0$ the loss-pertubed label $y^*_w(\epsilon)$ chooses a configuration with a higher loss and performs a gradient descent step on $\nabla_w\mu_w(x,y^*_w(\epsilon))$, i.e., it moves the parameters w to reduce the score function for the high-loss label $\mu_w(x,y^*_w(\epsilon))$. When $\epsilon<0$ the loss-perturbed label $y^*_w(\epsilon)$ chooses a configuration with a lower loss and performs a gradient descent step on $-\nabla_w\mu_w(x,y^*_w(\epsilon))$, i.e., it increases the score function for the low-loss label $\mu_w(x,y^*_w(\epsilon))$. This choice is especially important in structured prediction, when there might be exponentially many structures with high loss and only few structures with low loss. This observation already appears in the original direct loss work (Hazan et al., 2010) (see last paragraph of Section 2).

5.1. Bipartite Matchings

We follow the problem setting, architecture $\mu_{u,\alpha}(x,y_\alpha)$ and loss function $\ell(y,y^*)$ of Mena et al. (2018) for learning bipartite matching, and replace the Gumbel-Sinkhorn operation with our gradient step, see Figure 2.

In this experiment each training example $(x,y)\in S$ consists of an input vector $x\in R^d$ of d numbers drawn independently from the uniform distribution over the $[0,1]$ interval. The structured label y, $y\in\{0,1\}^{d^2}$, is a bipartite matching between the elements of x to the elements of the sorted vector of x. Formally, $y_{ij}=1$ if $x_i=\text{sort}(x)_j$ and zero otherwise. Here we set α to be the pair of indexes $i,j=1,\ldots,d$ that corresponds to the desired bipartite matching. The network learns a real valued number for each (i,j)-th entry, namely, $\mu_{u,ij}(x,y_{ij})$ and our gradient update rule in Equation (16) replaces the Gumbel-Sinkhorn operator of Mena et al. (2018). We note that $y^*_{w,\gamma}$ can be computed efficiently using any max-matching algorithm,

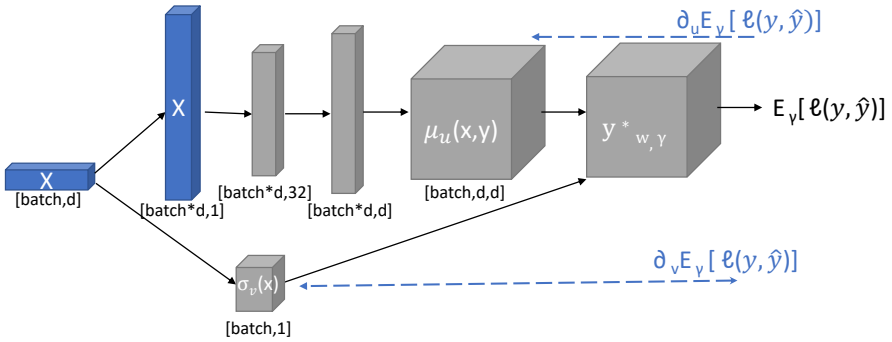


Figure 2. Architecture for learning bipartite matchings: The expectancy over Gumbel noise of the loss is derived w.r.t. the parameters u of the signal and w.r.t. the parameters v of the variance controller σ directly (Equations 16,17 respectively). The network μ has a first fully connected layer that links the sets of samples to an intermediate representation (with 32 neurons), and a second (fully connected) layer that turns those representations into batches of latent permutation matrices of dimension d by d each. It has the same architecture as the equivalent experiment by Mena et al. (2018). The network σ has a single layer connecting input sample sequences to a single output which is then activated by a softplus activation. We chose such an activation to enforce a positive σ value.

which maximizes a linear function over the set of possible matching Y: $y^*_{w,\gamma} =$

$$\arg\max_{\hat{y}\in Y} \sum_{ij=1}^{d} \mu_{u,ij}(x,\hat{y}_{ij}) + \sum_{ij=1}^{d} \sigma_v(x)\gamma_{ij}(\hat{y}_{ij}) \qquad (18)$$

Our gradient computation also requires the loss perturbed predictor $y^*_{w,\gamma}(\epsilon)$, which takes into account the quadratic loss function:

$$\ell(y,\hat{y}) = \sum_{i=1}^{d}\left(\sum_{j=1}^{d} x_j y_{ij} - \sum_{j=1}^{d} x_j \hat{y}_{ij}\right)^2$$

Note that this loss function is not smooth, as it is a function of binary elements, i.e. the squared differences between one hot vectors. Seemingly, the quadratic loss function does not decompose along the score structure $\mu_{u,ij}(x,y_{ij})$, therefore it is challenging to recover the loss-perturbed prediction efficiently. Instead, we use the fact that $y^2_{ij} = y_{ij}$ for $y_{ij} \in \{0,1\}$ and represent the loss as a linear function over the set of all matchings: $\ell(y,\hat{y}) = t + \sum_{j=1}^{d} t_{ij}\hat{y}_{ij}$, with $t = \sum_{ij=1}^{d} x_j^2 y_{ij}$ and $t_{ij} = \sum_{i=1}^{d} x_j^2(1-2y_{ij})$. (Further details in the supplementary material). With this, we are able to recover the loss-perturbed predictor $y^*_{w,\gamma}(\epsilon)$ with the same computational complexity as $y^*_{w,\gamma}$, i.e., using linear solver over maximum matching:

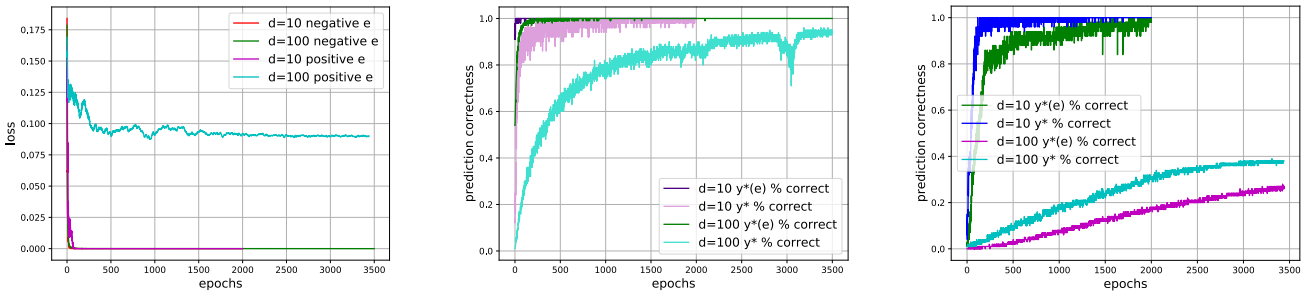
$$\begin{aligned} y^*_{w,\gamma}(\epsilon) \quad &= \quad \arg\max_{\hat{y}\in Y} \sum_{ij=1}^{d} \mu_{u,ij}(x,\hat{y}_{ij}) \qquad (19) \\ &+ \quad \sum_{ij=1}^{d} \sigma_v(x)\gamma_{ij}(\hat{y}_{ij}) + \epsilon \sum_{ij=1}^{d} t_{ij}\hat{y}_{ij}. \end{aligned}$$

Note that we may omit t from the optimization since it does not impact the maximal argument.

In our experimental validation we found that negative ϵ works the best. In this case, when $y_{ij} = 0$ the corresponding embedding potential $\mu_{u,ij}(x,y_{ij})$ is perturbed by $-|\epsilon|x_j^2$, while when $y_{ij} = 1$ it is increased by $|\epsilon|x_j^2$. Doing so incrementally pushes $y^*_{w,\gamma}(\epsilon)$ towards predicting the ground truth permutation, which is aligned with our intuition of the towards-best direct loss minimization. The dynamics of our method as a function of matching dimension under a positive versus a negative ϵ is illustrated in Figure 3. In Figure 3a we plot the loss as a function of training epochs with varying size of matching dimension d. While the loss is similar for $\epsilon > 0$ and $\epsilon < 0$ when $d = 10$, this changes for $d = 100$. As such, the percentage of correctly sorted input entries of y^*_w and $y^*(\epsilon)$ greatly differs when $d = 100$ for different ϵ. Importantly, when learning with $\epsilon > 0$ there are less than 40% correct entries in y^*_w (Figure 3c), while when learning with $\epsilon < 0$ there are at least 90% correct entries in y^*_w (Figure 3b).

Mena et al. (2018) have introduced two evaluation measures: the proportion of sequences where there was at least one error (Prop. Any Wrong), and the overall proportion of samples assigned to a wrong position (Prop. Wrong). They report the best achieved Prop. Any Wrong measure over an unspecified number of trials. To indicate robustness, we extend these measures to the following: Percentage of zero Prop. Any Wrong sequences, as well as Average and STD of Prop. Wrong, which are calculated over a number of training and testing repetitions.

We follow the Sorting Numbers experiment protocol of Mena et al. (2018) and use the code released by the authors, to perform 20 Sinkhorn iterations and 10 different reconstruction for each batch sample. Also, the training set consists of 10 random sequences of length d and a test set that consists of a single sequence of the same length d. At test time, random noise is not added to the learned sig-



(a) loss with negative and positive ϵ (b) y^* and $y^*(\epsilon)$ with negative ϵ (c) y^* and $y^*(\epsilon)$ with positive ϵ

Figure 3. The effect of the sign of ϵ, as a function of dimension during training. We plot the percentage of correctly sorted input entries of the predictor (y^*) and the loss-augmented predictor ($y^*(\epsilon)$) as well as the loss. While the loss is similar for negative and positive ϵ when $d = 10$, this changes for $d = 100$ (Figure 3a). As such, the percentage of correctly sorted input entries of y^*_w and $y^*(\epsilon)$ greatly differs when $d = 100$ for different ϵ. Importantly, when learning with $\epsilon > 0$ there are less than 40% correct entries in y^*_w (Figure 3c), while when learning with $\epsilon < 0$ there are at least 90% correct entries in y^*_w (Figure 3b).

nal $\mu_{u,ij}(x, y_{ij})$. The results in Table 1 show the measures calculated over 200 repetitions of training and testing.

One can see that direct loss minimization performs better than Gumbel-Sinkhorn, and the gap is larger for longer sequences. One can also see that learning the variance of the noise improves the performance of the structured predictor in all three measures, when compared to direct loss minimization (Hazan et al., 2010), in which the variance is set to zero, as well as to (Lorberbom et al., 2018), in which the noise variance is set to one.

Running time comparison is given in Table 2. Our code may be found in `https://github.com/HeddaCohenIndelman/PerturbedStructuredPredictorsDirect`.

5.2. k-Nearest Neighbors For Image Classification

We follow the problem setting and architecture $\mu_{u,\alpha}(x, y_\alpha)$ and loss function $\ell(y, y^*)$ of Grover et al. (2019) for learning the k-nearest neighbors (kNN) classier. We replace the unimodal row stochastic matrix operation with our gradient step to directly minimize the distance to the closest k candidates, see Figure 4. In this experiment each training example $(x, y) \in S$ consists of an input vector $x = (x_1, ..., x_n, x_q)$ of n candidate images $x_1, ..., x_n$ and a single query image x_q; its corresponding structured label $y = (y_1, ..., y_n)$. The structured label $y \in \{0, 1\}^n$ points to the k candidate images with minimum Euclidean distance to the query image, i.e., $\sum_{i=1}^n y_i = k$. Here we set α to be the index $i = 1, \ldots, n$ that correspond to the top k candidate images. $\mu_{u,i}(x, y_i)$ is the negative distance between the embedding $h_u(\cdot)$ of the i-th candidate image and that of the query image: $\mu_{u,i}(x, y_i) = -\|h_u(x_i) - h_u(x_q)\|$. Our prediction $y^*_{u,\gamma}$ yields the top-k images having the minimum Euclidean distance in embedding space (equivalently, the maximum negative Euclidean distance) $y^*_{u,k,\gamma} = \arg\max_{\hat{y}\in Y}\{\mu_u(x, \hat{y}) + \gamma(\hat{y})\}$. Here,

Table 1. Bipartite Matching Evaluation Measures. Results show Percentage of zero Prop. Any Wrong sequences of test set (i.e perfect sorting). Average and STD of Prop. Wrong in parenthesis. We show the effect of learning signal-to-noise ratio method 'Direct Stochastic Learning' in comparison with 'Direct $\bar{\sigma} = 0$' referring to direct loss minimization (Hazan et al., 2010), which can be interpreted as setting the noise variance to zero, 'Direct $\bar{\sigma} = 1$' referring to (Lorberbom et al., 2018), in which the noise variance is set to one, and 'Gumbel-Sinkhorn' referring to (Mena et al., 2018). Training set setting of 10 random sequences of length d and a test set of a single sequence of length d. Results are calculated from 200 training and testing repetitions.

d	Direct $\bar{\sigma} = 0$	Direct $\bar{\sigma} = 1$	**Direct Stochastic Learning**	Gumbel-Sinkhorn
5	98.5% (0.6%±4.9%)	100% (0%±0%)	**100% (0%±0%)**	100% (0%±0%)
10	97% (0.6%±3.4%)	100% (0%±0%)	**100% (0%±0%)**	100% (0%±0%)
25	89.5% (0.9%±2.8%)	97.5% (0.3%±1.7%)	**97.5% (0.3%±1.6%)**	87.5% (1%±3%)
40	84.5% (1.2%±4.5%)	90.5% (0.6%±2.2%)	**91.6% (0.5%±1.6%)**	83.5% (1%±5%)
60	82% (0.9%±2.6%)	80.0% (0.9%±2.2%)	**83.3% (0.7%±1.8%)**	21% (5%±9%)
100	74.9% (1.4%±6.9%)	68.5% (1.2%±2.4%)	**76.8% (0.9%±2.1%)**	0% (11.3%±11.2%)

Table 2. Comparison of average epoch time (seconds) of the bipartite matching experiment, per selected d

d	**Direct Stochastic Learning**	**Gumbel-Sinkhorn**
10	0.247	0.288
40	0.252	0.294
100	0.304	0.306

we set Y to be the set of all structures $y \in \{0, 1\}^n$ satisfying $\sum_{i=1}^n y_i = k$. The loss function is a linear function of its labels: $\ell(y, \hat{y}) = -\sum_{i=1}^n \|x_i - x_q\| y_i \hat{y}_i$. Our gradient update rule in Equation (16) replaces the unimodal row stochastic construction operator of Grover et al. (2019). We note that $y^*_{w,\gamma}$ and $y^*_{w,\gamma}(\epsilon)$ can be computed efficiently by extracting

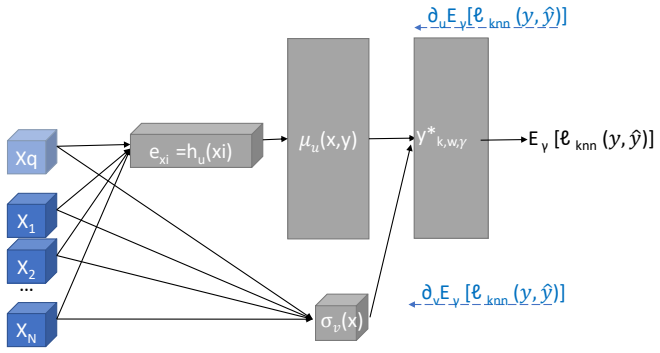


Figure 4. k-nn schematic architecture. The expectancy over Gumbel noise of the loss is derived w.r.t. the parameters u of the signal and w.r.t. the parameters v of the variance controller σ directly (Equations 16,17 respectively). We deployed the same distance embedding networks as the ones deployed by Grover et al. (2019) (Details in the supplementary material). Our prediction $y^*_{u,\gamma}$ yields the top-k images having the minimum Euclidean distance in embedding space (equivalently, the maximum negative Euclidean distance). The network σ output is activated by a softplus activation in order to enforce a positive σ value.

the top k elements over n elements.

Table 3. Test-Set Classification Average Accuracy, Per k.
We show the effect of our 'Direct Stochastic Learning' method for learning signal-to-noise ratio in comparison with: 'Direct $\bar{\sigma} = 0$' referring to direct loss minimization without random noise (Hazan et al., 2010), 'Direct $\bar{\sigma} = 1$' referring to Lorberbom et al. (2018), in which the noise variance is set to one, 'NeuralSort' referring to Grover et al. (2019), and 'RelaxSubSample' referring to Xie and Ermon (2019) who quote results for $k = 5$ only.

	MNIST			
	k=1	k=3	k=5	k=9
Direct $\bar{\sigma} = 0$	99.1%	99.2%	99.3%	99.2%
Direct $\bar{\sigma} = 1$	16%	53.84%	14.25%	41.53%
Direct Stochastic Learning	**99.34%**	99.4%	**99.4%**	99.34%
NeuralSort deterministic	99.2%	**99.5%**	99.3%	99.3%
NeuralSort stochastic	99.1%	99.3%	**99.4%**	**99.4%**
RelaxSubSample			99.3%	

	Fashion-MNIST			
	k=1	k=3	k=5	k=9
Direct $\bar{\sigma} = 0$	89.8%	93.2%	93.5%	**93.7%**
Direct $\bar{\sigma} = 1$	92.5%	**93.4%**	93.3%	93.2%
Direct Stochastic Learning	**92.6%**	93.3%	**94%**	**93.7%**
NeuralSort deterministic	**92.6%**	93.2%	93.5%	93%
NeuralSort stochastic	92.2%	93.1%	93.3%	93.4%
RelaxSubSample			93.6%	

	CIFAR-10			
	k=1	k=3	k=5	k=9
Direct $\bar{\sigma} = 0$	24.9%	27%	39.6%	39.9%
Direct $\bar{\sigma} = 1$	23.1%	89.95%	90.85%	91.6%
Direct Stochastic Learning	29.6%	**90.7%**	**91.25%**	**91.7%**
NeuralSort deterministic	**88.7%**	90.0%	90.2%	90.7%
NeuralSort stochastic	85.1%	87.8%	88.0%	89.5%
RelaxSubSample			90.1%	

We report the classification accuracies on the standard test sets in Table 3. For MNIST and Fashion-MNIST, our method matched or outperformed 'NeuralSort' (Grover et al., 2019) and 'RelaxSubSample' (Xie and Ermon, 2019), in all except $k = 3, 9$ in MNIST. For CIFAR-10, our method outperformed 'NeuralSort' and 'RelaxSubSample', in all except k=1, for which disappointing results are attained by all direct loss based methods. We note that 'Direct $\bar{\sigma} = 0$' seems to suffer from very low average accuracy on CIFAR-10 dataset. Additionally, 'Direct $\bar{\sigma} = 1$' suffer from very low average accuracy on MNIST dataset. It is evident that our method stabilizes the performance on all datasets.

Running time comparison for $k = 3$ is given in Table 4. Performance is robust to k in both methods.

Table 4. Comparison of average epoch running time (seconds) of the k-nn experiment. Results are for $k = 3$.

	Direct Stochastic Learning	NeuralSort Stochastic
MNIST	28.3	14.4
Fashion-MNIST	198.6	328.6
CIFAR-10	220.	337.6

6. Discussion And Future Work

In this work, we learn the mean and the variance of structured predictors, while directly minimizing their loss. Our work extends direct loss minimization as it explicitly adds random perturbation to the prediction process to better control the relation between data instance and its exponentially many possible structures. Our work also extends direct optimization through the $\arg\max$ in generative learning as it adds a variance term to better balance the learned signal with the perturbed noise. The experiments validate the benefit of our approach.

The structured distributions that are implied from our method are different than the standard Gibbs distribution, when the localized score functions are over subsets of variables. The exact relation between these distributions and the role of the Gumbel distribution law in the structured setting is an open problem. There are also optimization-related questions that arise from our work, such as exploring the role of ϵ and its impact on the convergence of the algorithm.

References

R. P. Adams and R. S. Zemel. Ranking via sinkhorn propagation. *ArXiv*, abs/1106.1925, 2011.

E. Andriyash, A. Vahdat, and W. G. Macready. Improved gradient-based optimization over discrete distributions. *ArXiv*, abs/1810.00116, 2018.

L. Berrada, A. Zisserman, and M. P. Kumar. Smooth loss functions for deep top-k classification. In *International Conference on Learning Representations*, 2018.

Q. Berthet, M. Blondel, O. Teboul, M. Cuturi, J.-P. Vert, and F. R. Bach. Learning with differentiable perturbed optimizers. *ArXiv*, abs/2002.08676, 2020.

M. Blondel, O. Teboul, Q. Berthet, and J. Djolonga. Fast differentiable sorting and ranking. In *ICML*, 2020.

J. Domke. Implicit differentiation by perturbation. In *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc., 2010.

A. Grover, E. Wang, A. Zweig, and S. Ermon. Stochastic optimization of sorting networks via continuous relaxations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1eSS3CcKX.

T. Hazan, J. Keshet, and D. A. McAllester. Direct loss minimization for structured prediction. In *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc., 2010.

E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *International Conference on Learning Representations*, 2016.

W. Jin, R. Barzilay, and T. Jaakkola. Hierarchical generation of molecular graphs using structural motifs. *ArXiv*, abs/2002.03230, 2020.

A. Kendall, V. Badrinarayanan, , and R. Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

J. Keshet, C.-C. Cheng, M. Stoehr, and D. A. McAllester. Direct error rate minimization of hidden markov models. In *INTERSPEECH*, 2011.

W. Kool, H. van Hoof, and M. Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 3499–3508. PMLR, 2019. URL http://dblp.uni-trier.de/db/conf/icml/icml2019.html#KoolHW19.

S. Linderman, G. Mena, H. Cooper, L. Paninski, and J. Cunningham. Reparameterizing the birkhoff polytope for variational permutation inference. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1618–1627, 2018.

G. Lorberbom, A. Gane, T. S. Jaakkola, and T. Hazan. Direct optimization through arg max for discrete variational auto-encoder. In *NeurIPS*, 2018.

G. Lorberbom, C. J. Maddison, N. M. O. Heess, T. Hazan, and D. Tarlow. Direct policy gradients: Direct optimization of policies in discrete action spaces. *ArXiv*, abs/1906.06062, 2019.

C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

E. G. Mena, D. Belanger, S. Linderman, and J. Snoek. Learning latent permutations with gumbel-sinkhorn networks. *International Conference on Learning Representations*, 2018.

V. Niculae, A. F. T. Martins, M. Blondel, and C. Cardie. Sparsemap: Differentiable sparse structured inference. In *ICML*, 2018.

M. B. Paulus, D. Choi, D. Tarlow, A. Krause, and C. J. Maddison. Gradient estimation with stochastic softmax tricks. *ArXiv*, abs/2006.08063, 2020.

T. Plötz and S. Roth. Neural nearest neighbors networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1087–1098. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7386-neural-nearest-neighbors-networks.pdf.

M. V. Pogancic, A. Paulus, V. Musil, G. Martius, and M. Rolinek. Differentiation of blackbox combinatorial solvers. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=BkevoJSYPB.

R. Santa Cruz, B. Fernando, A. Cherian, and S. Gould. Visual permutation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):3100–3114, 2019.

Y. Song, A. G. Schwing, R. S. Zemel, and R. Urtasun. Training deep neural networks via direct loss minimization. In *ICML*, 2016.

S. Wiseman and A. M. Rush. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1137. URL https://www.aclweb.org/anthology/D16-1137.

S. M. Xie and S. Ermon. Reparameterizable subset sampling via continuous relaxations. In S. Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3919–3925. ijcai.org, 2019. doi: 10.24963/ijcai.2019/544. URL https://doi.org/10.24963/ijcai.2019/544.

Y. Xie, H. Dai, M. Chen, B. Dai, T. Zhao, H. Zha, W. Wei, and T. Pfister. Differentiable top-k operator with optimal transport. *ArXiv*, abs/2002.06504, 2020.