Joint Learning of Energy-based Models and their Partition Function

Michaël E. Sander [1] **Vincent Roulet** [1] **Tianlin Liu** [1] **Mathieu Blondel** [1]

Abstract

Probabilistic energy-based models (EBMs) offer a flexible framework for parameterizing probability distributions using neural networks. However, learning probabilistic EBMs by exact maximum likelihood estimation (MLE) is generally intractable, due to the need to compute the partition function (normalization constant). In this paper, we propose a novel formulation for approximately learning probabilistic EBMs in combinatorially-large discrete spaces, such as sets or permutations. Our key idea is to jointly learn both an energy model and its log-partition, both parameterized as neural networks. Our approach not only provides a novel tractable objective criterion to learn EBMs by stochastic gradient descent (without relying on MCMC), but also a novel means to estimate the log-partition function on unseen data points. On the theoretical side, we show that our approach recovers the optimal MLE solution when optimizing in the space of continuous functions. Furthermore, we show that our approach naturally extends to the broader family of Fenchel-Young losses, allowing us to obtain the first tractable method for optimizing the sparsemax loss in combinatorially-large spaces. We demonstrate our approach on multilabel classification and label ranking.

1. Introduction

Probabilistic energy-based models (EBMs) are a powerful framework for parameterizing probability distributions using neural networks, without any factorization assumptions. They have been successfully used both in the unsupervised setting (density estimation, image generation), where the goal is to learn a probability density function $p(\boldsymbol{x})$ and in the supervised learning setting (classification, structured prediction), where the goal is to learn a conditional probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$. Probabilistic EBMs include Bolzmann machines (Ackley et al., 1985) as a special case.

Unfortunately, this flexibility makes learning, sampling and inference with probabilistic EBMs much more challenging. Indeed, without factorization assumptions, turning an energy function into a valid probability distribution involves computing the partition function (normalization constant), which is generally intractable to compute exactly in continuous spaces or in combinatorially-large discrete spaces. To perform maximum likelihood estimation (MLE) of the network parameters, a standard approach is to rely on Markov chain Monte Carlo (MCMC) methods to estimate stochastic gradients of the log-likelihood function (Song & Kingma, 2021). However, deriving an MCMC sampler is usually case by case. For instance, for continuous outputs we may use Langevin-MCMC and for binary outputs we may use Gibbs sampling. In practice, as is the case in contrastive divergences (Hinton, 2002), MCMC iterations are not run to convergence, leading to biased stochastic gradient estimates (Fischer & Igel, 2010). In continuous spaces, score matching (Hyvärinen & Dayan, 2005) is often used as an alternative to MLE but it requires computing the Laplacian operator, which is challenging in high dimension. In combinatorially-large discrete spaces, one can use the generalized perceptron loss (LeCun et al., 2006) or the generalized Fenchel-Young loss (Blondel et al., 2022), which only involve computing an argmax or a regularized argmax, respectively. These losses sidestep computing the partition function, but they learn non-probabilistic EBMs.

In this paper, we propose a novel formulation for learning probabilistic EBMs $p(\boldsymbol{y}|\boldsymbol{x})$ in combinatorially-large discrete spaces (such as sets or permutations in our experiments) by MLE or any loss function in the Fenchel-Young family (Blondel et al., 2020a). Our approach not only provides a new tractable objective criterion that can be optimized by stochastic gradient descent (SGD) without MCMC, but also a new means to estimate the log-partition function on unseen data points using a jointly-learned separate neural network. We demonstrate our method on multilabel classification (space of sets) and label ranking (space of permutations).

Contributions.

- After reviewing some extensive background, we propose

[1]Google DeepMind. Correspondence to: Mathieu Blondel <mblondel@google.com>, Michaël E. Sander <michaelsander@google.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

a novel min-min formulation for learning probabilistic EBMs $p(\boldsymbol{y}|\boldsymbol{x})$ in combinatorially-large discrete spaces. Our approach consists in jointly learning two separate functions: an energy model and its log-partition.

- When minimizing the proposed objective in the space of continuous functions, we show that our approach exactly recovers MLE.
- In practice, we propose to parameterize the log-partition as a neural network and we replace minimization in the space of functions with minimization in the space of network parameters. An advantage of our approach is that the learned log-partition network can benefit from the universal approximation properties of neural networks, as we empirically demonstrate.
- To jointly learn the energy model and the log-partition, we propose an MCMC-free doubly stochastic optimization scheme. Our stochastic gradients are unbiased: our only assumption is that we are able to sample from a reference distribution, which can be the uniform distribution when no prior knowledge is available.
- We generalize our approach to the broader family of Fenchel-Young losses (Blondel et al., 2020a), such as the sparsemax loss (Martins & Astudillo, 2016). In this case, our approach consists in jointly learning the energy-based model and a Lagrange mutiplier (dual variable), that we parametrize as a neural network in practice. This is to our knowledge the first tractable approach for learning EBMs with the sparsemax loss in general combinatorially-large discrete spaces.

Notation. We denote the continuous input space by $\mathcal{X}$ and the discrete output set by $\mathcal{Y}$. We denote the set of continuous functions $g\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ as $\mathcal{F}(\mathcal{X} \times \mathcal{Y})$. Similarly, we use $\mathcal{F}(\mathcal{X})$ for continuous functions $\tau\colon \mathcal{X} \to \mathbb{R}$. We denote the set of conditional positive measures over $\mathcal{Y}$ conditioned on $\mathcal{X}$ as $\mathcal{P}_+(\mathcal{Y}|\mathcal{X})$. Similarly, we use $\mathcal{P}_1(\mathcal{Y}|\mathcal{X})$ for conditional probability measures. Given $\boldsymbol{x} \in \mathcal{X}$, we denote the partial evaluation of $g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})$ as $g(\boldsymbol{x}, \cdot) \in \mathcal{F}(\mathcal{Y})$. Given $h \in \mathcal{F}(\mathcal{Y})$ and $p \in \mathcal{P}_+(\mathcal{Y})$, we define $\langle h, p\rangle := \sum_{\boldsymbol{y}\in\mathcal{Y}} h(\boldsymbol{y})p(\boldsymbol{y})$. We denote the convex conjugate of $f(u)$ by $f^*(v) := \sup_u uv - f(u)$. Throughout this paper, we will use $\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}$ as a shorthand for $\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\rho_{\mathcal{X}\times\mathcal{Y}}}$ where $\rho_{\mathcal{X}\times\mathcal{Y}}$ is the joint data distribution over $\mathcal{X} \times \mathcal{Y}$ and $\mathbb{E}_{\boldsymbol{x}}$ as a shorthand for $\mathbb{E}_{\boldsymbol{x}\sim\rho_{\mathcal{X}}}$, where $\rho_{\mathcal{X}}$ is the marginal data distribution, i.e., $\rho_{\mathcal{X}}(\boldsymbol{x}) := \sum_{\boldsymbol{y}\in\mathcal{Y}} \rho_{\mathcal{X}\times\mathcal{Y}}(\boldsymbol{x}, \boldsymbol{y})$.

2. Background

2.1. Probabilistic energy-based models (EBMs)

Given a function $g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})$, which captures the affinity between an input $\boldsymbol{x} \in \mathcal{X}$ and an output $\boldsymbol{y} \in \mathcal{Y}$ (typically, a parameterized neural network), we define a probabilistic energy-based model (EBM) as

$$p_g(\boldsymbol{y}|\boldsymbol{x}) := \frac{q(\boldsymbol{y}|\boldsymbol{x})\exp(g(\boldsymbol{x}, \boldsymbol{y}))}{\sum_{\boldsymbol{y}'\in\mathcal{Y}} q(\boldsymbol{y}'|\boldsymbol{x})\exp(g(\boldsymbol{x}, \boldsymbol{y}'))}, \tag{1}$$

where $q \in \mathcal{P}_+(\mathcal{Y}|\mathcal{X})$ is a prior conditional positive measure. When such a measure is not available, we can just use the uniform positive measure, which gives

$$p_g(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp(g(\boldsymbol{x}, \boldsymbol{y}))}{\sum_{\boldsymbol{y}'\in\mathcal{Y}} \exp(g(\boldsymbol{x}, \boldsymbol{y}'))}.$$

We focus on (1) for generality, though in our experiments we will use a uniform reference measure. Throughout this paper, we leave the dependence on q implicit, as it is a fixed reference measure. It is well-known (Boyd, 2004) that (1) can be written from a variational perspective as

$$p_g(\cdot|\boldsymbol{x}) = \operatorname*{argmax}_{p\in\mathcal{P}_1(\mathcal{Y})} \langle g(\boldsymbol{x}, \cdot), p\rangle - \mathrm{KL}(p, q(\cdot|\boldsymbol{x})). \tag{2}$$

The corresponding log-partition, a.k.a. log-sum-exp, is

$$\mathrm{LSE}_g(\boldsymbol{x}) := \log \sum_{\boldsymbol{y}'\in\mathcal{Y}} q(\boldsymbol{y}'|\boldsymbol{x})\exp(g(\boldsymbol{x}, \boldsymbol{y}')). \tag{3}$$

It is related to the log-likelihood of the pair $(\boldsymbol{x}, \boldsymbol{y})$ through

$$\log p_g(\boldsymbol{y}|\boldsymbol{x}) = g(\boldsymbol{x}, \boldsymbol{y}) - \mathrm{LSE}_g(\boldsymbol{x}) + \log q(\boldsymbol{y}|\boldsymbol{x}). \tag{4}$$

Similarly to (2), we have

$$\mathrm{LSE}_g(\boldsymbol{x}) = \max_{p\in\mathcal{P}_1(\mathcal{Y})} \langle g(\boldsymbol{x}, \cdot), p\rangle - \mathrm{KL}(p, q(\cdot|\boldsymbol{x})). \tag{5}$$

Architecture of EBMs. As g is an affinity function, it is both natural and standard to decompose it as

$$g(\boldsymbol{x}, \boldsymbol{y}) := \Phi(h(\boldsymbol{x}), \boldsymbol{y}), \tag{6}$$

where $\Phi(\boldsymbol{\theta}, \boldsymbol{y})$ is a coupling function and $\boldsymbol{\theta} := h(\boldsymbol{x})$ is a model function producing logits $\boldsymbol{\theta} \in \Theta$ (Blondel et al., 2022). Typically, h and Φ are designed on a per-task basis. The simplest example of coupling is the bilinear form

$$\Phi(\boldsymbol{\theta}, \boldsymbol{y}) := \langle \boldsymbol{\theta}, \boldsymbol{y}\rangle. \tag{7}$$

In this case, an EBM coincides with an exponential family distribution with natural parameters $\boldsymbol{\theta}$. We will define more advanced couplings in Section 4.

2.2. Inference with EBMs

Computing the mode. Predicting the most likely output associated with an input $\boldsymbol{x}$ according to an EBM corresponds to computing the mode of the learned probability distribution, defined as

$$\boldsymbol{y}^\star_g(\boldsymbol{x}) := \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}} p_g(\boldsymbol{y}|\boldsymbol{x}) = \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})\exp(g(\boldsymbol{x}, \boldsymbol{y})). \tag{8}$$

Unfortunately, when $\mathcal{Y}$ is a combinatorially-large discrete set, this problem is often intractable. A common approximation is to replace $\mathcal{Y}$ with a convex set $\mathcal{C} \supset \mathcal{Y}$. The tightest such set is $\mathcal{C} = \mathrm{conv}(\mathcal{Y})$, the convex hull of $\mathcal{Y}$. For example, in multilabel classification with k labels, the output space is the power set of $[k]$, which can be represented as $\mathcal{Y} = \{0, 1\}^k$ and whose convex hull is $\mathrm{conv}(\mathcal{Y}) = [0, 1]^k$.

With uniform prior q, assuming that $g \in \mathcal{F}(\mathcal{X} \times \mathcal{C})$, and not just $g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})$, we can solve the relaxed problem

$$\boldsymbol{x} \mapsto \operatorname*{argmax}_{\boldsymbol{\mu} \in \mathcal{C}} g(\boldsymbol{x}, \boldsymbol{\mu}) \approx \boldsymbol{y}^\star_g(\boldsymbol{x}), \tag{9}$$

where we used that $\exp$ is monotonically increasing. If the coupling $\Phi(\boldsymbol{\theta}, \boldsymbol{\mu})$ is concave in $\boldsymbol{\mu}$, this problem can be solved optimally in polynomial time. Typically, one can use a gradient-based solver to obtain an approximate solution in $\mathcal{C}$ and then use a rounding procedure to produce a prediction that belongs to $\mathcal{Y}$. If $\Phi(\boldsymbol{\theta}, \boldsymbol{\mu})$ is the bilinear coupling (7), then (9) is known as maximum a posteriori (MAP) inference (Wainwright & Jordan, 2008) when $\mathcal{C} = \mathrm{conv}(\mathcal{Y})$. According to the fundamental theorem of linear programming, a solution happens at one of the vertices of $\mathcal{C}$ and therefore $\boldsymbol{y}^\star_g(\boldsymbol{x}) \in \mathcal{Y}$ in this case.

Computing the mean. Another important problem is that of computing the conditional expectation

$$\boldsymbol{\mu}_g(\boldsymbol{x}) \coloneqq \mathbb{E}_{\boldsymbol{y} \sim p_g(\cdot|\boldsymbol{x})}[\boldsymbol{y}] = \sum_{\boldsymbol{y} \in \mathcal{Y}} p_g(\boldsymbol{y}|\boldsymbol{x}) \boldsymbol{y} \in \mathrm{conv}(\mathcal{Y}). \tag{10}$$

In the special case of the bilinear coupling (7), we have that $\boldsymbol{\mu}_g(\boldsymbol{x})$ is the gradient w.r.t. h of $\mathrm{LSE}_g(\boldsymbol{x})$. Moreover, (10) is known as marginal inference and $\mathrm{conv}(\mathcal{Y})$ is known as the marginal polytope, because, if $\mathcal{Y}$ uses a binary encoding (indicator function) of outputs, then $\boldsymbol{\mu}_g(\boldsymbol{x})$ contains marginal probabilities (Wainwright & Jordan, 2008). For example, if $\mathcal{Y} = \{0, 1\}^k$, then $[\boldsymbol{\mu}_g(\boldsymbol{x})]_j$ contains the marginal probability of label j given $\boldsymbol{x}$.

When using the bilinear coupling (7) and when the variables in $\mathcal{Y}$ form a tree, LSE_g and $\boldsymbol{\mu}_g$ can be computed by message-passing-like dynamic programming algorithms (Wainwright & Jordan, 2008; Blondel & Roulet, 2024). Efficient algorithms also exist for particular structures, such as spanning trees (Zmigrod et al., 2021). However, these algorithms no longer work for non-bilinear couplings. In addition, even in the case of bilinear couplings, computing LSE_g and $\boldsymbol{\mu}_g$ is intractable in general. For example, if $\mathcal{Y}$ is the space of permutations, then computing LSE_g and $\boldsymbol{\mu}_g$ is known to be #P-complete, making exact MLE intractable.

2.3. MLE of EBMs is intractable in general

The log-likelihood of an observed pair $(\boldsymbol{x}, \boldsymbol{y})$ is given in (4). Accordingly, the MLE objective, which corresponds to expected risk minimization with a logistic loss, is

$$\begin{aligned}\mathcal{L}_{\mathrm{MLE}}(g) &\coloneqq \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\left[-\log p_g(\boldsymbol{y}|\boldsymbol{x})\right] \\ &= \mathbb{E}_{\boldsymbol{x}}\left[\mathrm{LSE}_g(\boldsymbol{x})\right] - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\left[g(\boldsymbol{x},\boldsymbol{y}) - \log(q(\boldsymbol{y}|\boldsymbol{x}))\right].\end{aligned} \tag{11}$$

Unfortunately, when $\mathcal{Y}$ is a combinatorially-large discrete set or a continuous set, this objective is intractable due to the log-partition function over all possible outputs.

2.4. Approximate MLE of EBMs with MCMC

Let us denote by $g_{\boldsymbol{w}}$ a function with parameters $\boldsymbol{w}$. Then, the gradient can be computed by (Song & Kingma, 2021)

$$\begin{aligned}\nabla_{\boldsymbol{w}} \mathcal{L}_{\mathrm{MLE}}(g_{\boldsymbol{w}}) =& \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{y}' \sim p_{g_{\boldsymbol{w}}}(\cdot|\boldsymbol{x})}[\nabla_{\boldsymbol{w}} g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}')]\right] - \\ & \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\left[\nabla_{\boldsymbol{w}} g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y})\right].\end{aligned}$$

For completeness, see Appendix C.7 for a proof. Therefore, as long as we can draw samples from $p_{g_{\boldsymbol{w}}}(\cdot|\boldsymbol{x})$, we can have access to an unbiased Monte-Carlo estimate of the gradient, allowing us to perform optimization based on stochastic gradient descent. However, sampling from $p_{g_{\boldsymbol{w}}}(\cdot|\boldsymbol{x})$ is difficult. The literature has focused on Markov chain Monte Carlo (MCMC) methods, such as Metropolis-Hastings, Gibbs sampling and Langevin. Typically, running MCMC until convergence is expensive. Therefore, MCMC is usually run for a small number of iterations, as in contrastive divergences (Hinton, 2002). Unfortunately, truncated MCMC can lead to biased gradient updates that hurt the learning dynamics (Song & Kingma, 2021). There are methods for bias removal but they typically greatly increase the variance.

2.5. Non-MLE probabilistic approaches

Departing from MLE, we can learn probabilistic EBMs of the form (1) using score matching (Hyvärinen & Dayan, 2005). It trains a model to estimate the gradient of the log-density function of the data, known as the score function. However, score matching requires the log-density of the data distribution to be continuously differentiable and finite everywhere, which limits it to continuous output spaces. In addition, score matching requires to compute and differentiate the Laplacian operator (the trace of the Hessian), which is often challenging in high dimension. Even though some extensions to discrete spaces have been proposed, such as concrete score matching (Meng et al., 2022) and ratio matching (Hyvärinen, 2007), score matching is seldom used in discrete spaces.

Another non-MLE approach is noise contrastive estimation (NCE) (Pihlaja et al., 2010; Gutmann & Hyvärinen, 2010; Gutmann & Hirayama, 2011). It works by training a model to discriminate between true data samples and generated noise samples. However, the performance of NCE can be sensitive to the choice of the noise distribution. In addition, the ratio of noise samples to data samples can affect performance and finding the optimal ratio usually requires experimentation.

2.6. Loss functions for non-probabilistic EBMs

When the goal is not to learn the probabilistic model (1) but just the relaxed argmax (9), we can use the generalized perceptron loss (LeCun et al., 2006),

$$(\boldsymbol{x}, \boldsymbol{y}) \mapsto \max_{\boldsymbol{\mu} \in \mathcal{C}} g(\boldsymbol{x}, \boldsymbol{\mu}) - g(\boldsymbol{x}, \boldsymbol{y}).$$

To ensure uniqueness of the argmax, we can add regularization Ω in (9) to define the regularized relaxed problem

$$\boldsymbol{x} \mapsto \operatorname*{argmax}_{\boldsymbol{\mu} \in \mathcal{C}} g(\boldsymbol{x}, \boldsymbol{\mu}) - \Omega(\boldsymbol{\mu}).$$

The corresponding generalized Fenchel-Young loss (Blondel et al., 2022) is

$$(\boldsymbol{x}, \boldsymbol{y}) \mapsto \max_{\boldsymbol{\mu} \in \mathcal{C}} g(\boldsymbol{x}, \boldsymbol{\mu}) - \Omega(\boldsymbol{\mu}) - g(\boldsymbol{x}, \boldsymbol{y}) + \Omega(\boldsymbol{y}).$$

Crucially, the gradient w.r.t. the parameters of g can be computed easily using the envelope theorem. However, these losses do not learn probabilistic models. They only learn to make the (relaxed) argmax predict the correct output. See also the discussion on distribution-space vs. mean-space losses in Blondel et al. (2020a).

2.7. Log-partition as an optimization variable

The idea of treating the log-partition as an optimization variable, rather than as a quantity to compute or estimate, was also explored in several earlier works in the unsupervised setting (Wang et al., 2018; Arbel et al., 2021; Senetaire et al., 2025). Compared to these works, our paper provides new theoretical guarantees regarding MLE recovery when optimizing in function space (Propositions 1 and 2), extends to the broader Fenchel-Young losses (Proposition 4), focuses on the supervised structured prediction setting, and most importantly, parameterizes the input-dependent log-partition as a neural network with generalization ability (in the unsupervised setting, the log-partition is an input-independent scalar value, therefore parameterizing the log-partition as a neural network would not make sense).

3. Proposed approach

In this section, we first propose a novel min-min formulation for learning probabilistic EBMs by MLE. We then generalize it to the family of Fenchel-Young losses.

3.1. A min-min formulation for MLE in EBMs

Overview of the approach. The partition function (normalization constant) in (1) can equivalently be formulated as an (intractable) equality constraint $\sum_{\boldsymbol{y} \in \mathcal{Y}} p_g(\boldsymbol{y}|\boldsymbol{x}) = 1$ for all $\boldsymbol{x} \in \mathcal{X}$. To deal with that constraint, our key idea is to introduce a Lagrange multiplier (dual variable), that we treat as a separate function $\tau \in \mathcal{F}(\mathcal{X})$ to minimize over. The Lagrange multiplier exactly coincides with the log-partition function in the MLE (logistic loss) case. Our approach consists in jointly learning the energy function $g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})$ and the log-partition function $\tau \in \mathcal{F}(\mathcal{X})$. Using duality arguments in the space of functions (see Proposition 1 below), we arrive at the objective function

$$\mathcal{L}_{\mathrm{MLE}}(g, \tau) := \mathbb{E}_{\boldsymbol{x}} \left[L_{g,\tau}(\boldsymbol{x})\right] - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})} \left[g(\boldsymbol{x}, \boldsymbol{y})\right], \quad (12)$$

where

$$L_{g,\tau}(\boldsymbol{x}) := \tau(\boldsymbol{x}) + \sum_{\boldsymbol{y}' \in \mathcal{Y}} q(\boldsymbol{y}'|\boldsymbol{x}) \left(\exp(g(\boldsymbol{x}, \boldsymbol{y}') - \tau(\boldsymbol{x})) - 1\right). \quad (13)$$

The minimization problem in the space of functions is then

$$\min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \min_{\tau \in \mathcal{F}(\mathcal{X})} \mathcal{L}_{\mathrm{MLE}}(g, \tau).$$

Intuitively, by minimizing the proposed objective with respect to both g and τ, we are able to approximate the true log-partition function while simultaneously learning the energy function so as to fit the data. In practice, we parameterize the energy model as $g_{\boldsymbol{w}}$ and the log-partition as $\tau_{\boldsymbol{v}}$. The minimization problem in the space of parameters is then

$$\min_{\boldsymbol{w} \in \mathcal{W}} \min_{\boldsymbol{v} \in \mathcal{V}} \mathcal{L}_{\mathrm{MLE}}(g_{\boldsymbol{w}}, \tau_{\boldsymbol{v}}).$$

In our experiments in Section 4, we consider linear models, multilayer pecerptrons (MLPs) and residual networks (ResNets) for $g_{\boldsymbol{w}}$, and we consider constant models, MLPs, ResNets and input-convex neural networks (ICNNs) (Amos et al., 2017) for $\tau_{\boldsymbol{v}}$.

Doubly stochastic gradient descent. When q is a probability distribution (if such a distribution is not available, we simply use the uniform distribution), we can rewrite (13) as

$$L_{g,\tau}(\boldsymbol{x}) = \tau(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{y}' \sim q(\cdot|\boldsymbol{x})} \left[\exp(g(\boldsymbol{x}, \boldsymbol{y}') - \tau(\boldsymbol{x}))\right] - 1. \quad (14)$$

As a result, we can estimate $\mathcal{L}_{\mathrm{MLE}}(g, \tau)$ and its stochastic gradients, provided that we can sample from $q(\cdot|\boldsymbol{x})$ for any $\boldsymbol{x}$. This suggests a doubly stochastic scheme, in which we sample both $(\boldsymbol{x}, \boldsymbol{y})$ pairs from the data distribution $\rho_{\mathcal{X} \times \mathcal{Y}}$ and $\boldsymbol{y}'$ prior samples from the reference distribution $q(\cdot|\boldsymbol{x})$; see Algorithm 1. For simplicity, we group $\boldsymbol{w} \in \mathcal{W}$ and $\boldsymbol{v} \in \mathcal{V}$ as a tuple and optimize both blocks of parameters simultaneously using Adam (Kingma, 2014). Although not explored in this work, it would also be possible to perform alternating minimization w.r.t. $\boldsymbol{w} \in \mathcal{W}$ and $\boldsymbol{v} \in \mathcal{V}$.

Algorithm 1 Doubly stochastic objective value computation

Inputs: models g and τ, batch sizes B and B'
Draw $(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_B, \boldsymbol{y}_B) \overset{\text{i.i.d.}}{\sim} \rho_{\mathcal{X}\times\mathcal{Y}}$
Draw $\boldsymbol{y}'_{i,1}, \ldots, \boldsymbol{y}'_{i,B'} \overset{\text{i.i.d.}}{\sim} q(\cdot|\boldsymbol{x}_i)$ for $i \in [B]$
$\tilde{L}_{g,\tau}(\boldsymbol{x}_i) := \tau(\boldsymbol{x}_i) + \frac{1}{B'}\sum_{j=1}^{B'}(\exp(g(\boldsymbol{x}_i, \boldsymbol{y}'_{i,j}) - \tau(\boldsymbol{x}_i)) - 1)$
Output: $\frac{1}{B}\left(\sum_{i=1}^{B} \tilde{L}_{g,\tau}(\boldsymbol{x}_i) - g(\boldsymbol{x}_i, \boldsymbol{y}_i)\right)$

Equivalence with MLE. Although our proposed objective (12) appears quite different from the original MLE (expected risk) objective, we can show that they are in fact equivalent when minimizing over the space of functions.

Proposition 1. *Optimality of min-min, MLE case*

Suppose that for all $\boldsymbol{y} \in \mathcal{Y}$, $\boldsymbol{x} \mapsto q(\boldsymbol{y}|\boldsymbol{x})$ is continuous. Then, we have

$$\min_{g \in \mathcal{F}(\mathcal{X}\times\mathcal{Y})} \mathcal{L}_{\mathrm{MLE}}(g) = \min_{\substack{g \in \mathcal{F}(\mathcal{X}\times\mathcal{Y}) \\ \tau \in \mathcal{F}(\mathcal{X})}} \mathcal{L}_{\mathrm{MLE}}(g, \tau)$$

and for all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$

$$\mathrm{LSE}_{g^\star}(\boldsymbol{x}) = \tau^\star(\boldsymbol{x})$$
$$p_{g^\star}(\boldsymbol{y}|\boldsymbol{x}) = q(\boldsymbol{y}|\boldsymbol{x})\exp(g^\star(\boldsymbol{x}, \boldsymbol{y}) - \tau^\star(\boldsymbol{x})).$$

See Appendix C.2 for a proof. The optimality in Proposition 1 holds because we perform minimization in the space of continuous functions. If we parameterize τ as a neural network, and perform minimization in the space of parameters instead, our approach only performs approximate MLE. However, thanks to the universality of neural networks, minimization in the space of parameters should be close to minimization in the space of continuous functions, provided that the neural network used is sufficiently expressive. This is confirmed in Figure 1. The larger the number of $\boldsymbol{y}'$ samples we draw to estimate the expectation in (14), which corresponds to B' in Algorithm 1, the faster we converge to the exact MLE objective.

Generalization ability of the learned log-partition. Our approach not only provides a tractable objective criterion to learn EBMs by SGD, but also a novel means to estimate the log-partition function on unseen data points. Indeed, because we parameterize $\tau_{\boldsymbol{v}}$ as a neural network, we can evaluate it on new datapoints at inference time. Figure 2 empirically confirms the generalization ability of our learned log-partition function on unseen data points.

Learning the partition function, as opposed to computing it or estimating it, bears some similarity with learning the value function in reinforcement learning (Konda & Tsitsiklis, 1999; Schulman et al., 2017). Parameterizing dual variables as neural networks, as we do in this work, has also been explored in optimal transport (Seguy et al., 2018; Korotin et al., 2021) and in generative adversarial networks (Nowozin et al., 2016).

3.2. Finite sum setting

The MLE objective (11) corresponds to the *expected* risk minization setting, in which there are potentially infinitely-many $(\boldsymbol{x}, \boldsymbol{y})$ pairs. In practice, we often work in the *empirical* risk minimization or finite sum setting, in which we use a finite number of pairs $(\boldsymbol{x}_1, \boldsymbol{y}_1)$, ..., $(\boldsymbol{x}_n, \boldsymbol{y}_n)$. The classical MLE objective (11) then becomes (removing constants)

$$\widetilde{\mathcal{L}}_{\mathrm{MLE}}(g) := \frac{1}{n}\left(\sum_{i=1}^{n} \mathrm{LSE}_g(\boldsymbol{x}_i) - g(\boldsymbol{x}_i, \boldsymbol{y}_i)\right) \tag{15}$$

and our proposed objective (12) becomes

$$\widetilde{\mathcal{L}}_{\mathrm{MLE}}(g, \tau) := \frac{1}{n}\left(\sum_{i=1}^{n} L_{g,\tau}(\boldsymbol{x}_i) - g(\boldsymbol{x}_i, \boldsymbol{y}_i)\right). \tag{16}$$

In the finite sum setting, if we do not need τ to generalize to unseen $\boldsymbol{x}$ instances (recall that the log-partition is not needed at inference time if all we want to compute is the mode, i.e., the argmax prediction), then we can set $\tau_{\boldsymbol{v}}(\boldsymbol{x}_i) := v_i$, for $\boldsymbol{v} := (v_1, \ldots, v_n) \in \mathbb{R}^n$. Our approach then solves the finite-sum MLE objective optimally.

Proposition 2. *Optimality, finite sum setting*

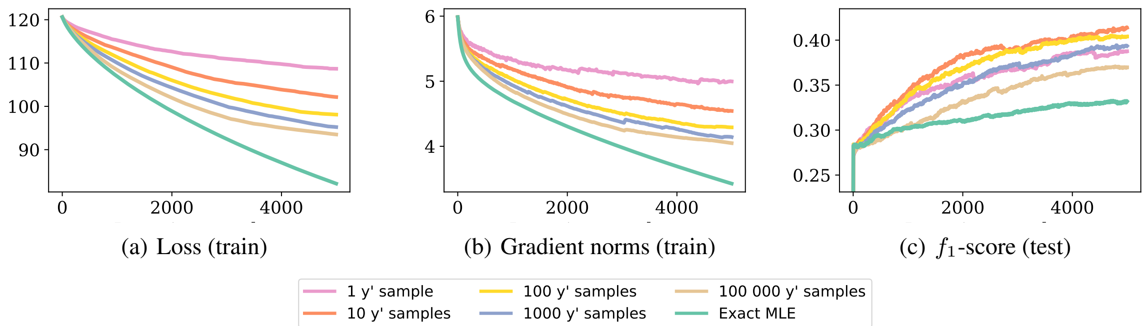
Suppose $\tau_{\boldsymbol{v}}(\boldsymbol{x}_i) := v_i$ for $i \in [n]$. Then,

$$\min_{\boldsymbol{w} \in \mathcal{W}} \widetilde{\mathcal{L}}_{\mathrm{MLE}}(g_{\boldsymbol{w}}) = \min_{\substack{\boldsymbol{w} \in \mathcal{W} \\ \boldsymbol{v} \in \mathbb{R}^n}} \widetilde{\mathcal{L}}_{\mathrm{MLE}}(g_{\boldsymbol{w}}, \tau_{\boldsymbol{v}})$$

and $v_i^\star = \mathrm{LSE}_{g_{\boldsymbol{w}^\star}}(\boldsymbol{x}_i)$ for all $i \in [n]$.

See Appendix C.3 for a proof. Remarkably, the result holds *even if* $g_{\boldsymbol{w}}$ is nonlinear in $\boldsymbol{w} \in \mathcal{W}$.

Joint convexity in the case of linear models. If $g_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \langle h_{\boldsymbol{w}}(\boldsymbol{x}_i), \boldsymbol{y}_i\rangle$, where $h_{\boldsymbol{w}}$ is linear in $\boldsymbol{w}$, then the classical MLE objective (15) is that of conditional random fields (Lafferty et al., 2001; Sutton & McCallum, 2012) and it is convex in $\boldsymbol{w}$. However, despite convexity, it could be intractable if for example $\mathcal{Y}$ is the set of permutations. In contrast, our objective (16) is jointly convex in $(\boldsymbol{w}, \boldsymbol{v})$, as proved in Appendix C.6, and it remains tractable as long as we can sample from $q(\cdot|\boldsymbol{x})$. In this convex setting, SGD enjoys a convergence rate of $O(1/\sqrt{t})$, and if we further add strongly-convex regularization on the parameters, the rate becomes $O(1/t)$ (Garrigos & Gower, 2023; Bach, 2024). Therefore, our objective can be optimized using stochastic gradient methods, and we have obtained convergence rates for learning EBMs in arbitrary combinatorial spaces.



(a) Loss (train) (b) Gradient norms (train) (c) f_1-score (test)

Figure 1. **Convergence of the proposed approach** as a function of the number of iterations, when varying the number of prior samples $\boldsymbol{y}'$ drawn. To be able to compute the exact MLE objective (15), we use the unary multilabel model (Section 4.1) on the cal500 dataset (174 classes and therefore 2^{174} possible configurations) as the test bed. Note that the loss and gradient in plots (a) and (b) are computed using (15) even for our method. We make two key observations: i) our approach converges to exact MLE as predicted by our theory, ii) the number of $\boldsymbol{y}'$ samples can have a regularization effect on the test f_1-score. See Figure 3 for more results.

3.3. Comparison with min-max formulation

One distinguishing feature of our approach is that it uses a min-min formulation. It is therefore insightful to compare it with a min-max formulation. With some overloading of the notation, let us define

$$\mathcal{L}_{\text{MLE}}(g,p) \coloneqq \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{y}'\sim p(\cdot|\boldsymbol{x})}[g(\boldsymbol{x},\boldsymbol{y}')] - \text{KL}(p(\cdot|\boldsymbol{x}), q(\cdot|\boldsymbol{x}))$$
$$- \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}[g(\boldsymbol{x},\boldsymbol{y})] + \text{const}.$$

We then have the following proposition.

Proposition 3. *Optimality of min-max, MLE case*

We have

$$\min_{g\in\mathcal{F}(\mathcal{X}\times\mathcal{Y})} \mathcal{L}_{\text{MLE}}(g) = \min_{g\in\mathcal{F}(\mathcal{X}\times\mathcal{Y})} \max_{p\in\mathcal{P}_1(\mathcal{Y}|\mathcal{X})} \mathcal{L}_{\text{MLE}}(g,p).$$

A proof is given in Appendix C.4. This is akin to adversarial approaches, where the function g plays the role of a discriminator and the distribution p plays the role of a generator (Nowozin et al., 2016; Ho & Ermon, 2016). Because the maximization w.r.t. p in the space of distributions is usually intractable when $\mathcal{Y}$ is combinatorial or infinite and because we need to be able to sample from $p(\cdot|\boldsymbol{x})$, it is common to replace p with a parameterized distribution from which it is easy to sample. An additional challenge comes from gradient computations: since p appears in the sampling, one typically uses the score function estimator (REINFORCE) to estimate the gradient w.r.t. the parameters of p. However, this estimator suffers from high variance. In contrast, in our min-min approach, the variable τ is a function, not a distribution and its gradients are easy to compute, since expectations are w.r.t. q, not p.

3.4. Generalization to Fenchel-Young losses

Another advantage of our proposed approach is that it naturally generalizes to any loss function in the Fenchel-Young family (Blondel et al., 2020a). It is well-known that the log-sum-exp can be interpreted as a "softmax". To obtain a more general notion of softmax, we replace the KL divergence in (5) with f-divergences (Csiszár, 1967; Ali & Silvey, 1966), which are defined when $\mathcal{Y}$ is a finite set as

$$D_f(p,q) \coloneqq \sum_{\boldsymbol{y}\in\mathcal{Y}} f(p(\boldsymbol{y})/q(\boldsymbol{y}))q(\boldsymbol{y}).$$

We assume that f is strictly convex and differentiable on $(0,+\infty)$. We can then define the f-softmax as

$$\text{softmax}_g^f(\boldsymbol{x}) \coloneqq \max_{p\in\mathcal{P}_1(\mathcal{Y})} \langle g(\boldsymbol{x},\cdot),p\rangle - D_f(p,q(\cdot|\boldsymbol{x})),$$

and the f-softargmax as

$$p_g^f(\cdot|\boldsymbol{x}) \coloneqq \underset{p\in\mathcal{P}_1(\mathcal{Y})}{\text{argmax}} \langle g(\boldsymbol{x},\cdot),p\rangle - D_f(p,q(\cdot|\boldsymbol{x})).$$

By analogy with the MLE objective in (11), we can then consider the objective

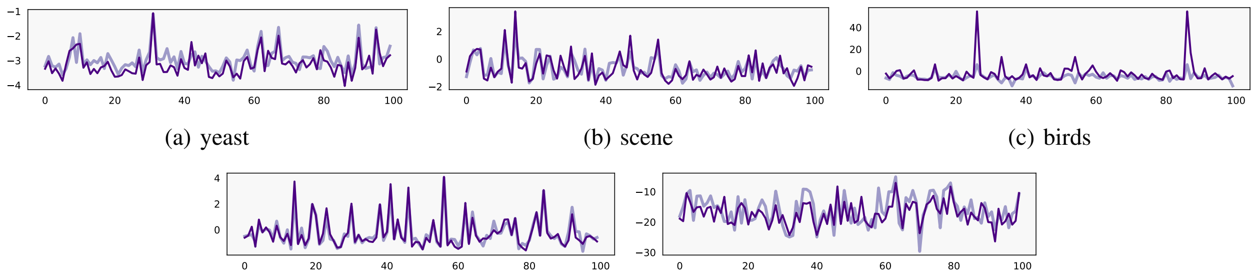
$$\mathcal{L}_f(g) \coloneqq \mathbb{E}_{\boldsymbol{x}}\left[\text{softmax}_g^f(\boldsymbol{x})\right] - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\left[g(\boldsymbol{x},\boldsymbol{y})\right].$$

Unfortunately, just as for the log-sum-exp in the MLE setting, the f-softmax is intractable to compute when $\mathcal{Y}$ is combinatorially large. Similarly to (12), using duality arguments, we arrive at the objective

$$\mathcal{L}_f(g,\tau) \coloneqq \mathbb{E}_{\boldsymbol{x}}\left[L_{g,\tau}^f(\boldsymbol{x})\right] - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\left[g(\boldsymbol{x},\boldsymbol{y})\right],$$

with $L_{g,\tau}^f(\boldsymbol{x}) \coloneqq \tau(\boldsymbol{x}) + \sum_{\boldsymbol{y}\in\mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) f_+^*(g(\boldsymbol{x},\boldsymbol{y}) - \tau(\boldsymbol{x}))$.

Here, f_+^* is the convex conjugate of f_+, the restriction of f to $\mathbb{R}_+$. We have the next generalization of Proposition 1.



(a) yeast (b) scene (c) birds

(d) emotions (e) cal500

Figure 2. **Generalization ability of the learned log-partition function for multilabel classification.** As a testbed to compare the learned log-partition (dark purple), we use the unary model (see Section 4.1 for details), which enjoys a closed-form expression for the exact log-partition (light purple). We pick randomly 100 test samples (x-axis) on 5 multilabel classification datasets, after training models with hyper-parameters optimized against the validation set.

Proposition 4. *Optimality of min-min, general case*

Let f be a strictly convex differentiable function such that $(0, +\infty) \subseteq \operatorname{dom} f'$. Suppose that for all $\boldsymbol{y} \in \mathcal{Y}$, $\boldsymbol{x} \mapsto q(\boldsymbol{y}|\boldsymbol{x})$ is continuous. Then, we have

$$\min_{g \in \mathcal{F}(\mathcal{X}\times\mathcal{Y})} \mathcal{L}_f(g) = \min_{\substack{g \in \mathcal{F}(\mathcal{X}\times\mathcal{Y}) \\ \tau \in \mathcal{F}(\mathcal{X})}} \mathcal{L}_f(g, \tau)$$

and for $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$

$$p^f_{g^\star}(\boldsymbol{y}|\boldsymbol{x}) = q(\boldsymbol{y}|\boldsymbol{x})(f^*_+)'(g^\star(\boldsymbol{x}, \boldsymbol{y}) - \tau^\star(\boldsymbol{x})).$$

A proof is given in Appendix C.5. In the MLE case, τ coincided exactly with the log-partition function. In the more general setting, τ corresponds to the Lagrange multiplier associated with the equality constraints $\sum_{\boldsymbol{y}\in\mathcal{Y}} p^f_g(\boldsymbol{y}|\boldsymbol{x}) = 1$.

Example: sparsemax for EBMs. When $f(u) = \frac{1}{2}(u^2 - 1)$, which is the generating function of the chi-square divergence, we obtain the first tractable method for optimizing the sparsemax loss (Martins & Astudillo, 2016) on EBMs in combinatorially-large discrete spaces. We have

$$L^f_{g,\tau}(\boldsymbol{x}) = \tau(\boldsymbol{x}) + \frac{1}{2} \sum_{\boldsymbol{y}\in\mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) \left[g(\boldsymbol{x}, \boldsymbol{y}) - \tau(\boldsymbol{x})\right]^2_+$$

and $p^f_{g^\star}(\boldsymbol{y}|\boldsymbol{x}) = q(\boldsymbol{y}|\boldsymbol{x}) \left[g^\star(\boldsymbol{x}, \boldsymbol{y}) - \tau^\star(\boldsymbol{x})\right]_+$, where $[\cdot]_+ := \max(., 0)$ is the non-negative part. Previous attempts to use the sparsemax loss for structured prediction require a k-best maximization oracle (Pillutla et al., 2018). Unfortunately, such oracles are not available for all sets $\mathcal{Y}$ (e.g., permutations) and their cost is usually polynomial in k, while k could be arbitrarily large. In contrast, our doubly-stochastic approach only requires us to be able to sample from the reference probability measure $q(\cdot|\boldsymbol{x})$.

4. Experiments

We now demonstrate our approach through experiments on multilabel classification and label ranking. Throughout our experiments, as in (6), we assume that $g(\boldsymbol{x}, \boldsymbol{y}) = \Phi(h(\boldsymbol{x}), \boldsymbol{y})$ where $\Phi(\boldsymbol{\theta}, \boldsymbol{y})$ is a coupling function and $h(\boldsymbol{x})$ is a model function producing logits $\boldsymbol{\theta}$. Experimental details and additional results are presented in Appendix A.

4.1. Multilabel classification

In this section, we perform experiments on multilabel classification with k classes. Here, given an input $\boldsymbol{x}$, the goal is to predict an output $\boldsymbol{y}$ belonging to the powerset of $[k]$, which can be represented as $\mathcal{Y} = \{0, 1\}^k$. The cardinality is $|\mathcal{Y}| = 2^k$ and the convex hull is $\operatorname{conv}(\mathcal{Y}) = [0, 1]^k$.

Unary model. To perform multilabel classification, the simplest approach is to use the bilinear coupling (7) $\Phi(\boldsymbol{\theta}, \boldsymbol{y}) = \sum_{j=1}^k \theta_j y_j$ together with logits $\boldsymbol{\theta} = h(\boldsymbol{x}) \in \mathbb{R}^k$. In this case, predicting the mode (8) with uniform q reads

$$\boldsymbol{y}^\star_g(\boldsymbol{x}) = \operatorname*{argmax}_{\boldsymbol{y}\in\{0,1\}^k} \Phi(\boldsymbol{\theta}, \boldsymbol{y}) = \operatorname*{argmax}_{\boldsymbol{y}\in\{0,1\}^k} \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle,$$

for which the optimal solution is

$$[\boldsymbol{y}^\star_g(\boldsymbol{x})]_j = \begin{cases} 1 & \text{if } \theta_j \geq 0 \\ 0 & \text{if } \theta_j < 0 \end{cases} \quad \forall j \in [k].$$

The log-sum-exp (3) and marginal inference (10) both enjoy closed-form solutions:

$$\operatorname{LSE}_g(\boldsymbol{x}) = \sum_{j=1}^k \operatorname{softplus}(\theta_j) \tag{17}$$
$$[\boldsymbol{\mu}_g(\boldsymbol{x})]_j = \operatorname{sigmoid}(\theta_j) \quad \forall j \in [k].$$

Therefore, the unary model boils down to placing sigmoid activations on top of the logits $\boldsymbol{\theta} = h(\boldsymbol{x})$, as usual in deep

Table 1. f_1-score on multi-label classification for different models and losses (with constant τ model, that is $\tau_{\boldsymbol{v}}(\boldsymbol{x}_i) \coloneqq v_i$.)

Loss	Model g	yeast	scene	birds	emotions	cal500
Logistic (min-min)	Linear (unary)	63.41	70.78	45.49	61.63	40.19
	MLP (unary)	65.04	75.35	**46.48**	65.33	44.91
	ResNet (unary)	65.03	75.64	36.93	**67.45**	45.23
	Linear (pairwise)	63.40	70.72	45.49	61.63	40.23
	MLP (pairwise)	**65.11**	75.22	46.08	64.84	45.00
	ResNet (pairwise)	64.84	75.68	41.34	64.26	44.61
Sparsemax (min-min)	Linear (unary)	62.92	70.43	45.12	63.14	45.18
	MLP (unary)	63.22	75.83	42.42	66.86	46.39
	ResNet (unary)	62.27	75.44	28.63	44.53	39.11
	Linear (pairwise)	62.91	69.89	45.12	63.14	45.18
	MLP (pairwise)	63.39	74.94	40.68	66.28	46.39
	ResNet (pairwise)	62.32	**75.63**	27.85	64.58	30.93
Logistic (exact MLE)	Linear (unary)	61.46	70.64	43.97	63.35	37.31
	MLP (unary)	60.71	70.81	40.88	62.99	36.24
	Resnet (unary)	62.27	71.02	40.78	61.50	37.45
Generalized Fenchel-Young	Linear (unary)	61.41	70.26	44.13	61.70	36.85
	MLP (unary)	61.31	70.84	40.73	61.85	37.35
	Linear (pairwise)	61.62	70.42	42.28	63.05	35.12
Logistic (MCMC)	Linear (unary)	61.98	70.53	44.93	61.76	44.61
	MLP (unary)	60.34	71.57	33.13	62.23	45.70
	Resnet (unary)	56.57	71.24	27.03	58.98	44.34
	Linear (pairwise)	61.96	70.50	44.93	61.76	44.61
	MLP (pairwise)	60.83	71.92	34.61	63.15	45.95
	Resnet (pairwise)	56.68	70.46	27.08	64.05	44.35
Logistic (min-max)	Linear (unary)	52.80	52.29	27.59	61.72	35.08
	MLP (unary)	58.58	52.54	24.95	60.53	46.08
	Resnet (unary)	57.53	42.82	25.32	55.85	47.59
	Linear (pairwise)	52.80	52.29	27.59	61.72	35.08
	MLP (pairwise)	58.76	51.59	26.71	58.68	43.31
	Resnet (pairwise)	57.53	42.95	22.35	46.72	**47.67**

learning. While our approach is not needed in the unary model, the unary model is a useful testbed for our method, as we can use (17) as a ground-truth for evaluating τ.

Pairwise model (Ising model). To try our approach on a model for which a closed form is not available for the log-partition, we consider a linear-quadratic coupling,

$$\Phi(\boldsymbol{\theta}, \boldsymbol{y}) \coloneqq \langle \boldsymbol{u}, \boldsymbol{y} \rangle + \frac{1}{2}\langle \boldsymbol{y}, \boldsymbol{U}\boldsymbol{y} \rangle = \sum_{j=1}^{k} u_j y_j + \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k} U_{i,j} y_i y_j,$$

where $\boldsymbol{\theta} \coloneqq (\boldsymbol{u}, \boldsymbol{U}) = h(\boldsymbol{x}) \in \mathbb{R}^k \times \mathbb{R}^{k \times k}$. As its name indicates, this coupling is linear in $\boldsymbol{\theta}$, but quadratic in $\boldsymbol{y}$. The model function h associates weights u_j to labels y_j and weights $U_{i,j}$ to pairwise label interactions $y_i y_j$. The relaxed inference problem (9) becomes

$$\boldsymbol{y}_g^\star(\boldsymbol{x}) \approx \operatorname*{argmax}_{\boldsymbol{\mu} \in [0,1]^k} \Phi(\boldsymbol{\theta}, \boldsymbol{\mu}) = \operatorname*{argmax}_{\boldsymbol{\mu} \in [0,1]^k} \langle \boldsymbol{u}, \boldsymbol{\mu} \rangle + \frac{1}{2}\langle \boldsymbol{\mu}, \boldsymbol{U}\boldsymbol{\mu} \rangle.$$

Contrary to the unary model case, there is no longer a closed-form solution. However, if we assume that $\boldsymbol{U}$ is negative semi-definite (which is easy to impose), this problem is concave in $\boldsymbol{\mu}$ and can be solved optimally without learning rate tuning using coordinate ascent (Blondel et al., 2022).

Results. We evaluate our models on classical multilabel classification datasets. The hyperparameters (learning rate and regularization) are optimized on a validation set. Once the best hyperparameters are found, we refit the model on the combined training and validation sets and evaluate it on the test set. Our results in Table 1 show that the logistic and sparsemax losses trained with our approach work better than the generalized Fenchel-Young loss as well as min-max and MCMC sampling approaches in various configurations. For the min-max approach, we use optimistic ADAM as solver, an MLP as generator and we use REINFORCE (score function estimator) for gradient estimation. For MCMC sampling, we use standard Metropolis–Hastings algorithm with uniform proposal distribution. We also present learning curves in Figure 1. We numerically validate that our approach converges to true MLE as more $\boldsymbol{y}'$ samples are used. We also observe a regularization phenomenon: using fewer $\boldsymbol{y}'$ samples leads to better f_1 scores than exact MLE. These findings are confirmed on other datasets (Figure 3).

Generalization ability of the learned partition function. A natural question is whether the learned log-partition τ generalizes to unseen test samples, that is, whether it correctly approximates the true log partition function. To empirically demonstrate this, we consider the unary model, for which the log-sum-exp enjoys the closed form solution (17). We parameterize τ as an MLP and g as a linear network. Af-

Table 2. Kendall rank correlation coefficient on label ranking for different models and losses, with MLP τ model

Loss	Polytope $\mathcal{C}$	Model g	authorship	glass	iris	vehicle	vowel	wine
Logistic (min-min)	$\mathcal{P}$	Linear	85.14	80.47	57.78	79.74	50.82	91.36
	$\mathcal{P}$	MLP	88.69	84.39	97.78	86.27	**74.26**	90.12
	$\mathcal{B}$	Linear	87.71	81.29	97.78	83.33	56.33	95.06
	$\mathcal{B}$	MLP	**90.80**	85.43	98.52	86.41	65.20	91.98
Sparsemax (min-min)	$\mathcal{P}$	Linear	84.88	79.84	52.59	78.76	50.25	93.83
	$\mathcal{P}$	MLP	86.52	**85.94**	**99.26**	85.75	68.34	91.36
	$\mathcal{B}$	Linear	89.09	79.43	97.78	82.81	56.24	90.12
	$\mathcal{B}$	MLP	87.84	80.47	98.52	**87.19**	55.43	85.80
Logistic (MCMC)	$\mathcal{P}$	Linear	84.29	80.67	55.56	79.15	49.97	91.36
	$\mathcal{P}$	MLP	87.90	75.50	97.04	86.34	57.90	91.98
	$\mathcal{B}$	Linear	88.10	79.33	97.78	82.22	55.99	**96.91**
	$\mathcal{B}$	MLP	89.68	80.16	97.78	85.88	48.90	91.98
Logistic (min-max)	$\mathcal{P}$	Linear	61.01	79.22	61.48	70.20	45.68	65.43
	$\mathcal{P}$	MLP	70.02	67.03	55.56	68.56	40.22	70.99
	$\mathcal{B}$	Linear	82.58	78.40	61.48	74.90	52.86	93.21
	$\mathcal{B}$	MLP	78.50	71.68	42.96	68.76	48.99	83.33

ter training, we evaluate the learned τ on 100 randomly sampled test samples and compare it with the true log-sum-exp. Results are shown in Figure 2. We observe strong correlation between the learned and true log-sum-exp.

4.2. Label ranking

In this section, we perform label ranking of k labels. Given an input $\boldsymbol{x}$, the goal is to predict a ranking $\boldsymbol{y}$ of the labels, that is, a permutation of $(1, \ldots, k)$. The cardinality is then $|\mathcal{Y}| = k!$. We focus on the bilinear coupling $\Phi(\boldsymbol{\theta}, \boldsymbol{y}) = \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle$. We consider two representations for permutations.

Permutahedron. A permutation can be represented as a vector. In this case, $\mathrm{conv}(\mathcal{Y}) = \mathcal{P}$, where $\mathcal{P}$ is the permutahedron (Bowman, 1972; Ziegler, 2012; Blondel et al., 2020b). We set the logits to $\boldsymbol{\theta} = h(\boldsymbol{x}) \in \mathbb{R}^k$. The argmax $\boldsymbol{y}^\star_g(\boldsymbol{x}) = \mathrm{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \Phi(\boldsymbol{\theta}, \boldsymbol{y}) = \mathrm{argmax}_{\boldsymbol{y} \in \mathcal{P}} \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle$ can be solved optimally using an argsort (Sander et al., 2023, Proposition 1). However, LSE_g and $\boldsymbol{\mu}_g$ are known to be #P-complete to compute, making classical MLE intractable.

Birkhoff polytope. A permutation can also be represented as a permutation matrix. In this case, $\mathrm{conv}(\mathcal{Y}) = \mathcal{B}$, where $\mathcal{B}$ is the Birkhoff polytope, the set of doubly-stochastic matrices. The logits are $\boldsymbol{\theta} = h(\boldsymbol{x}) \in \mathbb{R}^{k \times k}$. The argmax $\boldsymbol{y}^\star_g(\boldsymbol{x}) = \mathrm{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \Phi(\boldsymbol{\theta}, \boldsymbol{y}) = \mathrm{argmax}_{\boldsymbol{y} \in \mathcal{B}} \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle$ can be solved optimally using the Hungarian algorithm (Kuhn, 1955). However, LSE_g and μ_g are again #P-complete to compute, making classical MLE intractable.

Results. We use the same procedure as before to tune the hyperparameters. Our main results are in Table 2, confirming that our approach successfully learns in the space of permutations. The Birkhoff polytope works better than the permutahedron when using linear models. This makes sense because the logits when using the Birkhoff polytope are in $\mathbb{R}^{k \times k}$, enabling to capture label interactions even with a linear model, while the logits when using the permutahedron are in $\mathbb{R}^k$, which gives the ability to assign weight to single labels only. However, when using an MLP, we observe that the permutahedron works as well as the Birkhoff polytope, which is thanks to nonlinearity. The logistic and the sparsemax losses perform comparably overall.

5. Conclusion and future work

In this paper, we proposed a novel min-min formulation for jointly learning probabilistic EBMs and their log-partition in combinatorially-large spaces. Our method provably solves the MLE objective when minimizing the expected risk in the space of continuous functions. To optimize the model parameters, we proposed a doubly-stochastic MCMC-free SGD scheme, which only requires the ability to sample outputs from a prior reference probability distribution. We experimentally showed that parameterizing τ as a neural network leads to successful generalization on unseen data points. We demonstrated our method through experiments on multilabel and label ranking tasks.

In this paper, we evaluated our proposed method primarily in the prediction setting (finding the mode). Our paper does not directly address the challenge of sampling from $p(\boldsymbol{y}|\boldsymbol{x})$, which is central to evaluating performance in a probabilistic or generative setting. We leave to future work whether the learned log-partition function can be used to design better samplers.

Acknowledgements

We thank Alexandre Ramé as well as the anonymous reviewers for constructive feedbacks on this paper.

Impact statement

This paper introduces a novel formulation for learning conditional EBMs. We do not foresee any specific ethical or societal implications arising directly from this work.

References

Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9 (1):147–169, 1985.

Ali, S. M. and Silvey, S. D. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142, 1966.

Amos, B., Xu, L., and Kolter, J. Z. Input convex neural networks. In *Proceedings of the International Conference on Machine learning (ICML)*, pp. 146–155, 2017.

Arbel, M., Zhou, L., and Gretton, A. Generalized Energy Based Models. In *International Conference On Learning Representations*, 2021.

Bach, F. *Learning theory from first principles*. MIT press, 2024.

Blondel, M. and Roulet, V. The elements of differentiable programming. *arXiv preprint arXiv:2403.14606*, 2024.

Blondel, M., Martins, A. F., and Niculae, V. Learning with Fenchel–Young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020a.

Blondel, M., Teboul, O., Berthet, Q., and Djolonga, J. Fast differentiable sorting and ranking. In *Proceedings of the International Conference on Machine learning (ICML)*, pp. 950–959, 2020b.

Blondel, M., Llinares-López, F., Dadashi, R., Hussenot, L., and Geist, M. Learning energy networks with generalized Fenchel–young losses. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 35: 12516–12528, 2022.

Bowman, V. J. Permutation polyhedra. *SIAM Journal on Applied Mathematics*, 22(4):580–589, 1972.

Boyd, S. Convex optimization. *Cambridge UP*, 2004.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Csiszár, I. On information-type measure of difference of probability distributions and indirect observations. *Studia Sci. Math. Hungar.*, 2:299–318, 1967.

Fischer, A. and Igel, C. Empirical analysis of the divergence of gibbs sampling based learning algorithms for restricted boltzmann machines. In *International conference on artificial neural networks*, pp. 208–217. Springer, 2010.

Garrigos, G. and Gower, R. M. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

Gutmann, M. and Hirayama, J.-i. Bregman divergence as general framework to estimate unnormalized statistical models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 297–304, 2010.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

Ho, J. and Ermon, S. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Hyvärinen, A. Some extensions of score matching. *Computational Statistics & Data Analysis*, 51(5):2499–2512, 2007.

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

Kingma, D. P. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Konda, V. and Tsitsiklis, J. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Korotin, A., Li, L., Genevay, A., Solomon, J. M., Filippov, A., and Burnaev, E. Do neural optimal transport solvers work? a continuous wasserstein-2 benchmark. *Advances in neural information processing systems*, 34: 14593–14605, 2021.

Kuhn, H. W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2): 83–97, 1955.

Lafferty, J., McCallum, A., Pereira, F., et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine learning (ICML)*, 2001.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F., et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Martins, A. and Astudillo, R. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the International Conference on Machine learning (ICML)*, pp. 1614–1623, 2016.

Martins, A. F., Treviso, M., Farinhas, A., Aguiar, P. M., Figueiredo, M. A., Blondel, M., and Niculae, V. Sparse continuous distributions and Fenchel–Young losses. *Journal of Machine Learning Research*, 23(257):1–74, 2022.

Meng, C., Choi, K., Song, J., and Ermon, S. Concrete score matching: Generalized score matching for discrete data. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 34532–34545, 2022.

Mensch, A., Blondel, M., and Peyré, G. Geometric losses for distributional learning. In *Proceedings of the International Conference on Machine learning (ICML)*, pp. 4516–4525, 2019.

Nowozin, S., Cseke, B., and Tomioka, R. f-GAN: Training generative neural samplers using variational divergence minimization. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

Pihlaja, M., Gutmann, M., and Hyvärinen, A. A family of computationally efficient and simple estimators for unnormalized statistical models. In *Proc. Conf. on Uncertainty in Artificial Intelligence (UAI)*, pp. 442–449. AUAI Press, 2010.

Pillutla, V. K., Roulet, V., Kakade, S. M., and Harchaoui, Z. A smoother way to train structured prediction models. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 31, 2018.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 36, 2024.

Sander, M. E., Puigcerver, J., Djolonga, J., Peyré, G., and Blondel, M. Fast, differentiable and sparse top-k: a convex analysis perspective. In *Proceedings of the International Conference on Machine learning (ICML)*, pp. 29919–29936, 2023.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Seguy, V., Damodaran, B. B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. Large-scale optimal transport and mapping estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Senetaire, H., Jeha, P., Mattei, P.-A., and Frellsen, J. Learning Energy-Based Models by Self-normalising the Likelihood. *arXiv preprint arXiv:2503.07021*, 2025.

Song, Y. and Kingma, D. P. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Sutton, C. and McCallum, A. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373, 2012.

Wainwright, M. J. and Jordan, M. I. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Wang, B., Ou, Z., and Tan, Z. Learning Trans-dimensional Random Fields with Applications to Language Modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):876–890, 2018.

Ziegler, G. M. *Lectures on polytopes*, volume 152. Springer Science & Business Media, 2012.

Zmigrod, R., Vieira, T., and Cotterell, R. Efficient computation of expectations under spanning tree distributions. *Transactions of the Association for Computational Linguistics*, 9:675–690, 2021.

A. Experimental details and additional results

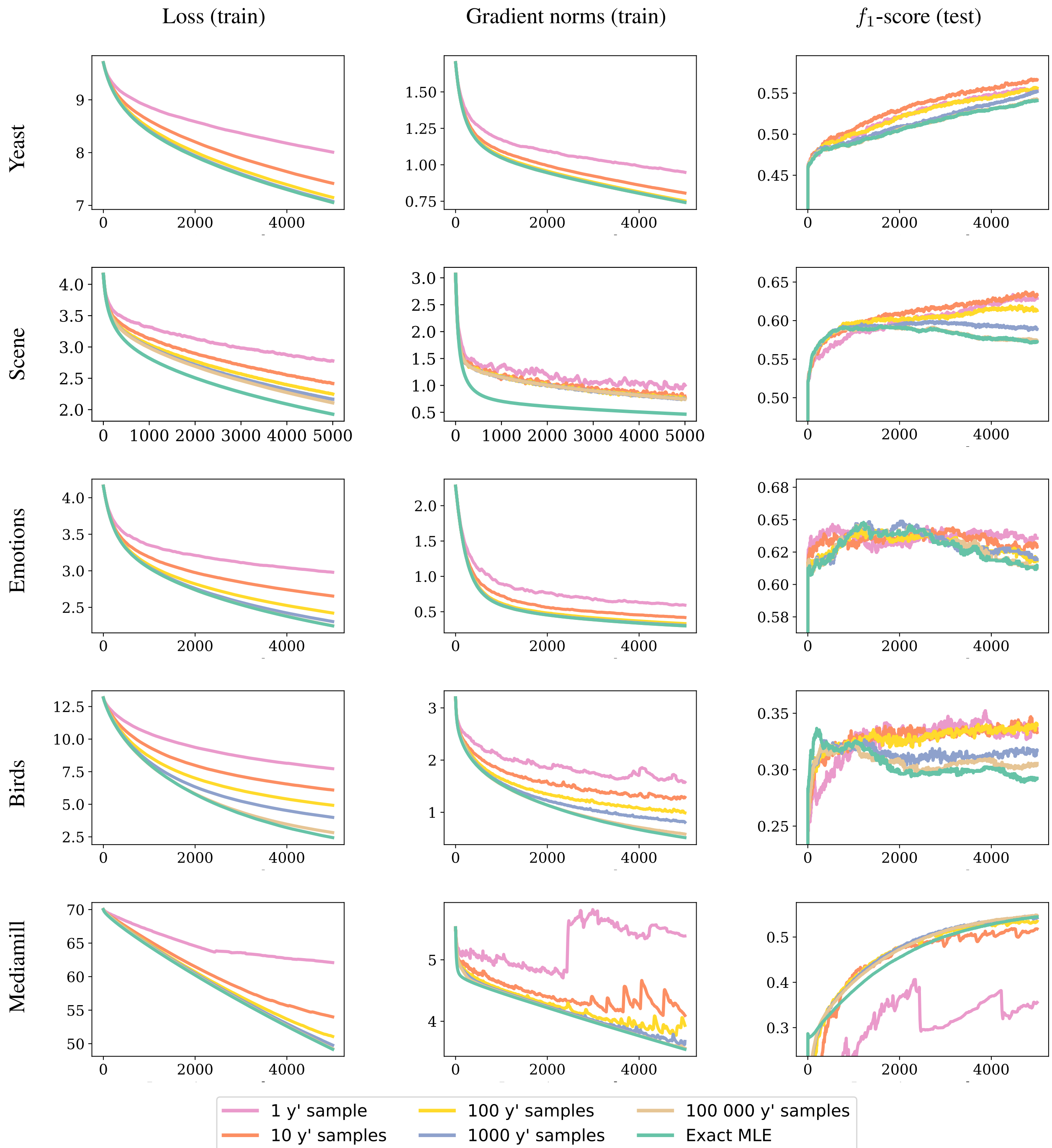


Figure 3. **Convergence of the proposed approach as a function of the number of prior samples y' drawn.** To be able to compute the exact MLE objective (15), we use the unary multilabel model (Section 4.1) as the test bed. Note that the loss and gradient in the left and center columns are computed using (15) even for our method. We make two key observations: i) our approach converges to exact MLE as predicted by our theory, ii) the number of y' samples can have a regularization effect on the test f_1-score.

Our implementation is made using JAX (Bradbury et al., 2018).

Multilabel classification datasets. We use the same datasets as in Blondel et al. (2022). The datasets can be downloaded from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. The dataset characteristics are described in Table 3 below.

Table 3. Dataset Characteristics

Dataset	Type	Train	Dev	Test	Features	Classes	Avg. labels
Birds	Audio	134	45	172	260	19	1.96
Cal500	Music	376	126	101	68	174	25.98
Emotions	Music	293	98	202	72	6	1.82
Mediamill	Video	22,353	7,451	12,373	120	101	4.54
Scene	Images	908	303	1,196	294	6	1.06
Yeast	Micro-array	1,125	375	917	103	14	4.17

Convergence curves. Convergence curves are in Figure 3. We use a linear model for g (unary model), and an MLP for τ with ReLU acivation and a hidden dimension of 128. Models are trained with the logistic loss. We use the Adam optimizer (Kingma, 2014) with a learning rate of 10^{-4} for the parameters of both g and τ. The models are trained for 5000 steps with full batch w.r.t. $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ pairs. The reason why we use full batches is to reduce the noise in the gradients and obtain a more stable training procedure.

Results over different models and losses. Additional experimental results are shown in Tables 4 and 5.

Table 4. f_1-score on multi-label classification for different models and losses, with ICNN τ model

Loss	Model g	yeast	scene	birds	emotions	cal500
logistic	Linear (unary)	63.53	70.44	45.36	62.90	42.74
	MLP (unary)	65.16	74.64	**46.58**	64.99	44.89
	Resnet (unary)	**65.40**	75.60	38.71	67.05	45.23
	Linear (pairwise)	63.53	70.47	45.36	62.90	42.75
	MLP (pairwise)	64.95	74.84	44.11	64.52	45.00
	Resnet (pairwise)	65.15	75.53	39.59	66.17	44.62
sparsemax	Linear (unary)	63.15	71.07	45.79	63.36	45.94
	MLP (unary)	63.52	74.34	40.77	**66.50**	46.44
	Resnet (unary)	62.62	**76.79**	30.18	63.38	**46.64**
	Linear (pairwise)	63.15	71.05	45.79	63.36	45.94
	MLP (pairwise)	63.25	74.93	41.61	66.18	46.40
	Resnet (pairwise)	62.07	75.59	28.14	63.42	37.53

Table 5. f_1-score on multi-label classification for different models, with logistic loss

Model τ	Model g	yeast	scene	birds	emotions	cal500
Constant	Linear (unary)	63.41	70.78	**45.49**	61.63	40.19
	MLP (unary)	65.04	75.35	46.48	65.33	44.91
	ResNet (unary)	65.03	**75.64**	36.93	**67.45**	45.23
	Linear (pairwise)	63.40	70.72	**45.49**	61.63	40.23
	MLP (pairwise)	65.11	75.22	46.08	64.84	45.00
	ResNet (pairwise)	64.84	75.68	41.34	64.26	44.61
MLP	Linear (unary)	61.51	70.77	44.65	60.93	42.40
	MLP (unary)	62.03	70.81	39.70	62.26	46.24
	ResNet (unary)	63.01	70.69	25.25	62.60	45.65
	Linear (pairwise)	61.52	70.81	44.65	60.85	42.08
	MLP (pairwise)	62.42	69.96	38.89	63.38	**46.08**
	ResNet (pairwise)	63.04	70.90	33.79	63.11	45.25
ICNN	Linear (unary)	63.53	70.44	45.36	62.90	42.42
	MLP (unary)	65.15	74.65	44.99	64.99	44.89
	ResNet (unary)	**65.39**	75.60	38.84	67.10	45.23
	Linear (pairwise)	63.53	70.47	45.36	62.90	42.42
	MLP (pairwise)	64.94	74.84	45.82	64.52	45.00
	ResNet (pairwise)	65.14	75.30	39.52	66.30	44.62

Label ranking. The publicly-available datasets can be downloaded from `https://github.com/akorba/Structured_Approach_Label_Ranking`. The experimental results are summarized in Tables 6 and 7.

Table 6. Kendall's tau on label ranking for different models and losses, with MLP τ model

Loss	Polytope $\mathcal{C}$	Model g	authorship	glass	iris	vehicle	vowel	wine
logistic	$\mathcal{P}$	Linear	85.14	80.47	57.78	79.74	50.82	91.36
logistic	$\mathcal{P}$	MLP	88.69	84.39	97.78	86.27	**74.26**	90.12
logistic	$\mathcal{P}$	Resnet	88.49	84.29	98.52	**87.06**	71.08	93.21
logistic	$\mathcal{B}$	Linear	87.71	81.29	97.78	83.33	56.33	**95.06**
logistic	$\mathcal{B}$	MLP	**90.80**	85.43	98.52	86.41	65.20	91.98
logistic	$\mathcal{B}$	Resnet	88.89	85.74	97.04	86.73	62.17	91.98
sparsemax	$\mathcal{P}$	Linear	84.88	79.84	52.59	78.76	50.25	93.83
sparsemax	$\mathcal{P}$	MLP	86.52	**85.94**	**99.26**	85.75	68.34	91.36
sparsemax	$\mathcal{P}$	Resnet	88.17	85.53	58.52	86.21	59.52	90.74
sparsemax	$\mathcal{B}$	Linear	89.09	79.43	97.78	82.81	56.24	90.12
sparsemax	$\mathcal{B}$	MLP	87.84	80.47	98.52	87.19	55.43	85.80
sparsemax	$\mathcal{B}$	Resnet	89.22	84.70	97.04	87.52	55.87	91.98

Table 7. Kendall's tau on label ranking for different models, with logistic loss

Polytope $\mathcal{C}$	Model τ	Model g	authorship	glass	iris	vehicle	vowel	wine
$\mathcal{P}$	Constant	Linear	84.94	79.64	57.78	79.61	49.18	91.98
$\mathcal{P}$	Constant	MLP	87.38	84.39	97.04	85.75	73.68	89.51
$\mathcal{P}$	Constant	ResNet	88.10	83.88	**98.52**	**87.19**	68.98	94.44
$\mathcal{B}$	Constant	Linear	88.17	77.16	97.78	83.66	57.29	95.06
$\mathcal{B}$	Constant	MLP	89.41	84.19	**98.52**	87.32	62.36	91.98
$\mathcal{B}$	Constant	ResNet	88.76	84.60	**98.52**	87.78	62.70	91.98
$\mathcal{P}$	MLP	Linear	85.14	80.47	57.78	79.74	50.82	91.36
$\mathcal{P}$	MLP	MLP	88.69	84.39	97.78	86.27	**74.26**	90.12
$\mathcal{P}$	MLP	Resnet	88.49	84.29	**98.52**	87.06	71.08	93.21
$\mathcal{B}$	MLP	Linear	87.71	81.29	97.78	83.33	56.33	**95.06**
$\mathcal{B}$	MLP	MLP	**90.80**	85.43	**98.52**	86.41	65.20	91.98
$\mathcal{B}$	MLP	Resnet	88.89	**85.74**	97.04	86.73	62.17	91.98
$\mathcal{P}$	ICNN	Linear	84.35	75.30	56.30	79.74	50.84	91.36
$\mathcal{P}$	ICNN	MLP	84.62	83.67	82.96	84.84	65.24	93.83
$\mathcal{P}$	ICNN	ResNet	87.25	82.53	85.19	83.66	65.09	93.83
$\mathcal{B}$	ICNN	Linear	88.76	81.71	87.41	82.88	53.94	**95.06**
$\mathcal{B}$	ICNN	MLP	90.20	77.98	90.37	84.97	46.83	94.44
$\mathcal{B}$	ICNN	ResNet	89.81	85.01	91.11	85.42	54.48	93.83

B. Additional materials

B.1. Loss functions using other forms of supervision

To workaround the intractable log-partition function arising in MLE, another alternative is to use loss functions that leverage other forms of supervision than $(\boldsymbol{x}, \boldsymbol{y})$ pairs. We review them for completeness, though we emphasize that these methods do *not* learn the probabilistic model (1).

Learning from pairwise preferences. If we have a dataset of triplets of the form $(\boldsymbol{x}, \boldsymbol{y}_+, \boldsymbol{y}_-)$ such that $\boldsymbol{y}_+ \succ \boldsymbol{y}_-$ given $\boldsymbol{x}$, we can learn a Bradley-Terry model

$$
\begin{aligned}
p_g(\boldsymbol{y}_1 \succ \boldsymbol{y}_2 | \boldsymbol{x}) &\coloneqq \operatorname{sigmoid}(g(\boldsymbol{x}, \boldsymbol{y}_1) - g(\boldsymbol{x}, \boldsymbol{y}_2)) \\
&= \frac{\exp(g(\boldsymbol{x}, \boldsymbol{y}_1))}{\exp(g(\boldsymbol{x}, \boldsymbol{y}_1)) + \exp(g(\boldsymbol{x}, \boldsymbol{y}_2))},
\end{aligned}
$$

where $\operatorname{sigmoid}(u) \coloneqq 1/(1 + \exp(u))$ is the logistic sigmoid. The loss associated with the triplet $(\boldsymbol{x}, \boldsymbol{y}_+, \boldsymbol{y}_-)$ is the pairwise logistic loss, $-\log p_g(\boldsymbol{y}_+ \succ \boldsymbol{y}_- | \boldsymbol{x})$. A major advantage of this approach is that there is no log-partition involved. The DPO loss (Rafailov et al., 2024) can be thought as a generalization of this pairwise logistic loss to non-uniform prior distributions q. However, it learns a probabilistic model of $\boldsymbol{y}_1 \succ \boldsymbol{y}_2$ given $\boldsymbol{x}$, not of $\boldsymbol{y}$ given $\boldsymbol{x}$.

Learning from pointwise scores. As an alternative, if we have a dataset of triplets $(\boldsymbol{x}, \boldsymbol{y}, t)$ where t is a scalar score assessing the affinity between $\boldsymbol{x}$ and $\boldsymbol{y}$, for instance a binary score or a real value score, we can use $\ell(g(\boldsymbol{x}, \boldsymbol{y}), t)$ for some classification or regression loss ℓ. Again, no log-partition is involved as we can estimate the parameters of g directly. However, this corresponds to learning a probabilistic model of t given $(\boldsymbol{x}, \boldsymbol{y})$, not of $\boldsymbol{y}$ given $\boldsymbol{x}$.

B.2. Fenchel-Young losses

Let us define some dual regularization

$$
\Omega_1(p) \coloneqq \Omega_+(p) + \iota_{\mathcal{P}_1(\mathcal{Y})}(p),
$$

where $\iota_{\mathcal{S}}$ is the indicator function of the set $\mathcal{S}$ and $\operatorname{dom}(\Omega_+) \subseteq \mathcal{P}_+(\mathcal{Y})$. A common choice for Ω_+ is the negative Shannon entropy $\Omega_+(p) = \langle p, \log p \rangle$ but we will see below how to define very general family of regularizations using f-divergences.

Given $g \in \mathcal{F}(\mathcal{Y})$ and $p \in \mathcal{P}_1(\mathcal{Y})$, we define the Fenchel-Young loss (Blondel et al., 2020a) regularized by Ω_1 as

$$\begin{aligned} L_{\Omega_1}(g, p) &\coloneqq \Omega_1^*(g) + \Omega_1(p) - \langle g, p \rangle \\ &= \Omega_1^*(g) + \Omega_1(p) - \mathbb{E}_{y \sim p}[g(y)]. \end{aligned}$$

The Fenchel-Young loss is primal-dual, in the sense that we see $g \in \mathcal{F}(\mathcal{Y})$ as a primal variable (a function) and $p \in \mathcal{P}_1(\mathcal{Y})$ as a dual variable (a distribution). We will omit the constant term $\Omega(p)$ in our derivations. While we focus on discrete spaces in this paper, Fenchel-Young losses were also extended to continuous spaces in Mensch et al. (2019); Martins et al. (2022).

C. Proofs

We define the shorthand notations $\rho_{\boldsymbol{x}} \coloneqq \rho(\cdot|\boldsymbol{x})$, $q_{\boldsymbol{x}} \coloneqq q(\cdot|\boldsymbol{x})$ and $g_{\boldsymbol{x}} \coloneqq g(\boldsymbol{x}, \cdot)$.

C.1. Lemmas

In this section, we state and prove useful lemmas.

Lemma 1. *Ω_1^* as unconstrained minimization of Ω_+^**

Let $\Omega_1 \coloneqq \Omega_+ + \iota_{\mathcal{P}_1(\mathcal{Y})}$, where Ω_+ is convex with $\operatorname{dom}(\Omega_+) \subseteq \mathcal{P}_+(\mathcal{Y})$. Then, for all $h \in \mathcal{F}(\mathcal{Y})$,

$$\Omega_1^*(h) = \min_{\tau \in \mathbb{R}} \tau + \Omega_+^*(h - \tau) \in \mathbb{R}$$

and

$$\nabla \Omega_1^*(h) = \nabla \Omega_+^*(h - \tau^\star) \in \mathcal{F}(\mathcal{Y}),$$

where $\tau^\star$ is an optimal solution.

Proof.

$$\begin{aligned} \Omega_1^*(h) &= \max_{p \in \mathcal{P}_1(\mathcal{Y})} \langle h, p \rangle - \Omega_+(p) \\ &= \max_{p \in \mathcal{P}_+(\mathcal{Y})} \min_{\tau \in \mathbb{R}} \langle h, p \rangle - \Omega_+(p) - \tau(\langle p, \mathbf{1} \rangle - 1) \\ &= \min_{\tau \in \mathbb{R}} \tau + \max_{p \in \mathcal{P}_+(\mathcal{Y})} \langle h - \tau, p \rangle - \Omega_+(p) \\ &= \min_{\tau \in \mathbb{R}} \tau + \Omega_+^*(h - \tau). \end{aligned}$$

The expression of $\nabla \Omega_1^*(h)$ follows from Danskin's theorem. □

Lemma 2. *Given $f_+ \colon \mathbb{R}_+ \to \mathbb{R}$, $p \in \mathcal{P}_+(\mathcal{Y})$ and a fixed reference measure $q \in \mathcal{P}_+(\mathcal{Y})$, let*

$$\Omega_+(p; q) \coloneqq \langle f_+(p/q), q \rangle = \sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}) f_+(p(\boldsymbol{y})/q(\boldsymbol{y})).$$

Then,

$$\Omega_+^*(h) = \sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}) f_+^*(h(\boldsymbol{y})) \in \mathbb{R}$$

and

$$\nabla \Omega_+^*(h)(\boldsymbol{y}) = q(\boldsymbol{y})(f_+^*)'(h(\boldsymbol{y})) \in \mathbb{R}_+.$$

Proof. This follows from classical conjugate calculus. □

C.2. Proof of Proposition 1 (equivalence with MLE, expected risk setting)

Proposition 1 is a corollary of Proposition 4 with the choice

$$f(u) \coloneqq u \log u - (u - 1) = f_+(u). \tag{18}$$

In this case, we obtain

$$f^*(v) = f_+^*(v) = \exp(v) - 1.$$

Using Lemma 2, we obtain for all $\boldsymbol{x} \in \mathcal{X}$

$$\Omega_+^*(g(\boldsymbol{x}, \cdot) - \tau(\boldsymbol{x}); q_{\boldsymbol{x}}) = \sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})[\exp(g(\boldsymbol{x}, \boldsymbol{y}) - \tau(\boldsymbol{x})) - 1].$$

For $\boldsymbol{x}$ fixed, minimizing w.r.t. τ, we get

$$\begin{aligned}
\sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})(f_+^*)'(g(\boldsymbol{x}, \boldsymbol{y}) - \tau(\boldsymbol{x})) = 1 &\iff \sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) \exp(g(\boldsymbol{x}, \boldsymbol{y}) - \tau(x)) = 1 \\
&\iff \sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) \exp(g(\boldsymbol{x}, \boldsymbol{y})) = \exp(\tau(\boldsymbol{x})) \\
&\iff \log \sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) \exp(g(\boldsymbol{x}, \boldsymbol{y})) = \tau(\boldsymbol{x}),
\end{aligned}$$

which is continuous in $\boldsymbol{x}$ if g is itself continuous. Thus the overall objective is

$$\min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \min_{\tau \in \mathcal{F}(\mathcal{X})} \mathbb{E}_{\boldsymbol{x}} \left[\tau(\boldsymbol{x}) + \sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})[\exp(g(\boldsymbol{x}, \boldsymbol{y}) - \tau(\boldsymbol{x})) - 1] \right] - \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y})} g(\boldsymbol{x}, \boldsymbol{y})$$

and

$$\pi_{g^\star, q}(\boldsymbol{y}|\boldsymbol{x}) = q(\boldsymbol{y}|\boldsymbol{x}) \exp(g^\star(\boldsymbol{x}, \boldsymbol{y}) - \tau(\boldsymbol{x}))$$

C.3. Proof of Proposition 2 (optimality in the empirical risk setting)

Using Lemma 1 with $h \coloneqq g(\boldsymbol{x}, \cdot)$, we obtain

$$\begin{aligned}
\min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \frac{1}{n} \sum_{i=1}^n [\Omega_1^*(g_{\boldsymbol{x}_i}; q_{\boldsymbol{x}_i}) - \langle g_{\boldsymbol{x}_i}, \delta_{\boldsymbol{y}_i} \rangle] &= \min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \frac{1}{n} \sum_{i=1}^n \left[\left(\min_{\tau_i \in \mathbb{R}} \tau_i + \Omega_+^*(g_{\boldsymbol{x}_i} - \tau_i; q_{\boldsymbol{x}_i}) \right) - \langle g_{\boldsymbol{x}_i}, \delta_{\boldsymbol{y}_i} \rangle \right]. \\
&= \min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \min_{\boldsymbol{\tau} \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \left[\left(\tau_i + \Omega_+^*(g_{\boldsymbol{x}_i} - \tau_i; q_{\boldsymbol{x}_i}) \right) - \langle g_{\boldsymbol{x}_i}, \delta_{\boldsymbol{y}_i} \rangle \right],
\end{aligned}$$

where $\delta_{\boldsymbol{y}} \in \mathcal{P}_1(\mathcal{Y})$ is a delta distribution such that $\delta_{\boldsymbol{y}}(\boldsymbol{y}') = 1$ for $\boldsymbol{y}' = \boldsymbol{y}$ and 0 for $\boldsymbol{y}' \neq \boldsymbol{y}$. Choosing (18) as in the proof of Proposition 1 gives the final result.

C.4. Proof of Proposition 3 (min-max formulation)

We have

$$\begin{aligned}
\min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \mathcal{L}(g) &= \min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \mathbb{E}_{\boldsymbol{x}} [\mathrm{LSE}_g(x) - \langle g_{\boldsymbol{x}}, \rho_{\boldsymbol{x}} \rangle] + \mathrm{const} \\
&= \min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \mathbb{E}_{\boldsymbol{x}} \left[\left(\max_{p_{\boldsymbol{x}} \in \mathcal{P}_1(\mathcal{Y})} \langle g_{\boldsymbol{x}}, p_{\boldsymbol{x}} \rangle - \mathrm{KL}(p_{\boldsymbol{x}}, q_{\boldsymbol{x}}) \right) - \langle g_{\boldsymbol{x}}, \rho_{\boldsymbol{x}} \rangle \right] + \mathrm{const} \\
&= \min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \max_{p \in \mathcal{P}_1(\mathcal{Y}|\mathcal{X})} \mathbb{E}_{\boldsymbol{x}} [(\langle g_{\boldsymbol{x}}, p_{\boldsymbol{x}} \rangle - \mathrm{KL}(p_{\boldsymbol{x}}, q_{\boldsymbol{x}})) - \langle g_{\boldsymbol{x}}, \rho_{\boldsymbol{x}} \rangle] + \mathrm{const} \\
&= \min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \max_{p \in \mathcal{P}_1(\mathcal{Y}|\mathcal{X})} \mathbb{E}_{\boldsymbol{x}} [(\mathbb{E}_{\boldsymbol{y} \sim p_{\boldsymbol{x}}}[g(\boldsymbol{x}, \boldsymbol{y})] - \mathrm{KL}(p_{\boldsymbol{x}}, q_{\boldsymbol{x}})) - \mathbb{E}_{\boldsymbol{y} \sim \rho_{\boldsymbol{x}}}[g(\boldsymbol{x}, \boldsymbol{y})]] + \mathrm{const}.
\end{aligned}$$

C.5. Proof of Proposition 4 (optimality in the general Fenchel-Young loss and expected risk setting)

Using Lemma 1 with $h \coloneqq g(\boldsymbol{x}, \cdot)$, we obtain

$$\min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \mathbb{E}_{\boldsymbol{x}} [\Omega_1^*(g_{\boldsymbol{x}}; q_{\boldsymbol{x}}) - \langle g_{\boldsymbol{x}}, \rho_{\boldsymbol{x}} \rangle] = \min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \mathbb{E}_{\boldsymbol{x}} \left[\left(\min_{\tau_{\boldsymbol{x}} \in \mathbb{R}} \tau_{\boldsymbol{x}} + \Omega_+^*(g_{\boldsymbol{x}} - \tau_{\boldsymbol{x}}; q_{\boldsymbol{x}}) \right) - \langle g_{\boldsymbol{x}}, \rho_{\boldsymbol{x}} \rangle \right].$$

We are going to show that

$$\min_{\tau_{\boldsymbol{x}} \in \mathbb{R}} \tau_{\boldsymbol{x}} + \Omega_+^*(g_{\boldsymbol{x}} - \tau_{\boldsymbol{x}}; q_{\boldsymbol{x}}) = \tau^\star(\boldsymbol{x}) + \Omega_+^*(g_{\boldsymbol{x}} - \tau^\star(\boldsymbol{x}); q_{\boldsymbol{x}})$$

where $\tau^\star \in \mathcal{F}(\mathcal{X})$.

Step 1. We first show that we can define a function $\boldsymbol{x} \mapsto \tau^\star(\boldsymbol{x})$.

We have $\tau_{\boldsymbol{x}} + \Omega_+^*(g_{\boldsymbol{x}} - \tau_{\boldsymbol{x}}; q_{\boldsymbol{x}}) = \tau_{\boldsymbol{x}} + \sum_{y \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) f^*(\max\{g(\boldsymbol{x}, \boldsymbol{y}) - \tau_{\boldsymbol{x}}, f'(0)\})$ with $f'(0) := \lim_{x \to 0, x \geq 0} f'(x) \in \mathbb{R} \cup \{-\infty\}$. Without loss of generality, we can assume that $q(\boldsymbol{y}|\boldsymbol{x}) > 0$ for all $\boldsymbol{y}$ and $\boldsymbol{x}$.

Since $(0, +\infty) \subseteq \operatorname{dom} f'$, $q > 0$ and $\mathcal{Y}$ is finite, $f'\left(\left(\sum_{y \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})\right)^{-1}\right)$ and $f'(\frac{1}{\min_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})})$ are well defined. We can then define

$$\begin{aligned}
\tau_{\min}(\boldsymbol{x}) &:= \max_{\boldsymbol{y} \in \mathcal{Y}} g(\boldsymbol{x}, \boldsymbol{y}) - f'\left(\frac{1}{\min_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})}\right) \\
\tau_{\max}(\boldsymbol{x}) &:= \max_{\boldsymbol{y} \in \mathcal{Y}} g(\boldsymbol{x}, \boldsymbol{y}) - f'\left(\left(\sum_{y \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})\right)^{-1}\right).
\end{aligned}$$

Since $q > 0$ and since f' is increasing, one has $\tau_{\min}(\boldsymbol{x}) < \tau_{\max}(\boldsymbol{x})$.

We can then analyze the following function on $[\tau_{\min}(\boldsymbol{x}), \tau_{\max}(\boldsymbol{x})] \times \mathcal{X}$:

$$h(\tau, \boldsymbol{x}) := \tau + \sum_{y \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) f^*(\max\{g(\boldsymbol{x}, \boldsymbol{y}) - \tau, f'(0)\}).$$

First, we need to ensure that we can compute derivatives of this function with respect to its first variable τ in $[\tau_{\min}(\boldsymbol{x}), \tau_{\max}(\boldsymbol{x})]$. We have that $f^*(\max\{g(\boldsymbol{x}, \boldsymbol{y}) - \tau, f'(0)\}) = (f + \iota_{\mathbb{R}_+})^*(g(\boldsymbol{x}, \boldsymbol{y}) - \tau)$ and $\operatorname{dom}(f + \iota_{\mathbb{R}_+})'_* = \operatorname{dom} f'_* \cup (-\infty, f'(0)]$. Therefore, if $f'(0) > -\infty$, the domain of $(f + \iota_{\mathbb{R}_+})'_*$ is unbounded below. Otherwise, if $f'(0) = -\infty$, since $\operatorname{Im} f' \subseteq \operatorname{dom} f'_*$, the domain of f'_*, and so of $(f + \iota_{\mathbb{R}_+})'_*$, are unbounded below. Denoting then $\alpha = \sup \operatorname{dom}(f + \iota_{\mathbb{R}_+})'_*$, since $\max_{\boldsymbol{y} \in \mathcal{Y}} g(\boldsymbol{x}, \boldsymbol{y}) - \tau_{\min}(\boldsymbol{x}) = f'(\frac{1}{\min_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})}) \in \operatorname{dom} f'_*$, we have $\max_{\boldsymbol{y} \in \mathcal{Y}} g(\boldsymbol{x}, \boldsymbol{y}) - \tau_{\min}(\boldsymbol{x}) < \alpha$ and therefore

$$\begin{aligned}
& \tau \geq \tau_{\min}(\boldsymbol{x}) \\
\Longrightarrow & \max_{\boldsymbol{y} \in \mathcal{Y}} g(\boldsymbol{x}, \boldsymbol{y}) - \tau < \alpha \\
\Longleftrightarrow & g(\boldsymbol{x}, \boldsymbol{y}) - \tau < \alpha, \text{for all } \boldsymbol{y} \in \mathcal{Y} \\
\Longrightarrow & \tau \in \operatorname{dom} h'.
\end{aligned}$$

We can then show that $h'(\tau_{\min}(\boldsymbol{x}), \boldsymbol{x}) \leq 0$ and $h'(\tau_{\max}(\boldsymbol{x}), \boldsymbol{x}) \geq 0$. Indeed, denoting $\boldsymbol{y}^\star \in \operatorname{argmax} g(\boldsymbol{x}, \boldsymbol{y})$ (we omit the dependency in x to ease the notations), we have

$$\begin{aligned}
h'(\tau_{\min}(\boldsymbol{x}), \boldsymbol{x}) &= 1 - \sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) f'_*(\max\{g(\boldsymbol{x}, \boldsymbol{y}) - \tau_{\min}(\boldsymbol{x}), f'(0)\}) \\
&\overset{(i)}{\leq} 1 - q(\boldsymbol{y}^\star|\boldsymbol{x}) f'_*(\max\{g(\boldsymbol{x}, \boldsymbol{y}^\star) - \tau_{\min}(\boldsymbol{x}), f'(0)\}) \\
&= 1 - q(\boldsymbol{y}^\star|\boldsymbol{x}) f'_*(\max\{f'(\frac{1}{\min_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})}), f'(0)\}) \\
&\overset{(ii)}{=} 1 - \frac{q(\boldsymbol{y}^\star|\boldsymbol{x})}{\min_{y \in \mathcal{Y}}(q(\boldsymbol{y}, \boldsymbol{x}))} \leq 0,
\end{aligned}$$

where in (i) we used that $q > 0$ and $f'_*(\max\{z, f'(0)\}) \geq 0$ for any $z \in \operatorname{dom}(f + \iota_{\mathbb{R}_+})'_*$, and in (ii), we used that f' is increasing and $f'_*(f'(p)) = p$ for any $p \in \operatorname{dom} f'$.

Similarly, we have

$$\begin{aligned}h'(\tau_{\max}(\boldsymbol{x}), \boldsymbol{x}) &= 1 - \textstyle\sum_{y \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) f'_*(\max\{g(\boldsymbol{x}, \boldsymbol{y}) - \tau_{\max}(\boldsymbol{x}), f'(0)\}) \\ &\overset{(i)}{\geq} 1 - \left(\textstyle\sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})\right) f'_*(\max\{g(\boldsymbol{x}, \boldsymbol{y}^\star) - \tau_{\max}, f'(0)\}) \\ &= 1 - \left(\textstyle\sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})\right) f'_* \left(\max\left\{f'\left(\left(\textstyle\sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x})\right)^{-1}\right), f'(0)\right\}\right) \\ &\overset{(ii)}{=} 0,\end{aligned}$$

where in (i) we used that $\sum_{\boldsymbol{y}} a_{\boldsymbol{y}} b_{\boldsymbol{y}} \leq (\sum_{\boldsymbol{y}} a_{\boldsymbol{y}}) \max_{\boldsymbol{y}} b_{\boldsymbol{y}}$ if $a_{\boldsymbol{y}} \geq 0$ with here $a_{\boldsymbol{y}} = q(\boldsymbol{y}|\boldsymbol{x}) > 0$ and $b_{\boldsymbol{y}} = f'_*(\max\{g(\boldsymbol{x}, \boldsymbol{y}) - \tau_{\max}(\boldsymbol{x}), f'(0)\})$, and in (ii) we used the same reasoning as for $h'(\tau_{\min})$.

Finally, we show that h' is increasing on $[\tau_{\min}(\boldsymbol{x}), \tau_{\max}(\boldsymbol{x})]$. When $\tau \leq \tau_{\max}(\boldsymbol{x})$, one has $\max_{\boldsymbol{y} \in \mathcal{Y}} g(\boldsymbol{x}, \boldsymbol{y}) - \tau \geq \max_{\boldsymbol{y} \in \mathcal{Y}} g(\boldsymbol{x}, \boldsymbol{y}) - \tau_{\max}(x) \geq f'(0)$.

Since $\theta \mapsto f'_*(\max\{\theta, f'(0)\})$ is increasing on $\operatorname{dom} f^* \setminus (-\infty, f'(0)]$ we have that $\tau \mapsto -f'_*(\max\{(g(\boldsymbol{x}, \boldsymbol{y}^\star) - \tau, f'(0)\})$ is increasing on $[\tau_{\min}(\boldsymbol{x}), \tau_{\max}(\boldsymbol{x})]$ and therefore $\tau \mapsto h'(\tau, \boldsymbol{x}) = 1 - \sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) f'_*(\max\{g(\boldsymbol{x}, \boldsymbol{y}) - \tau, f'(0)\})$ is increasing on $[\tau_{\min}(\boldsymbol{x}), \tau_{\max}(\boldsymbol{x})]$.

Overall h is well defined and strictly convex on $[\tau_{\min}(\boldsymbol{x}), \tau_{\max}(\boldsymbol{x})]$ such that $h'(\tau_{\min}(\boldsymbol{x})) \leq 0$ and $h'(\tau_{\max}(\boldsymbol{x})) \geq 0$. Hence, since h is convex, we have

$$\inf_{\tau \in \mathbb{R}} h(\tau, \boldsymbol{x}) = \min_{\tau_{\min}(\boldsymbol{x}) \leq \tau \leq \tau_{\max}(\boldsymbol{x})} h(\tau, \boldsymbol{x})$$

and the unique minimizer $\tau^\star(\boldsymbol{x})$ can then be found by solving the first order optimality condition $h'(\tau, \boldsymbol{x}) = 0$ in $[\tau_{\min}(\boldsymbol{x}), \tau_{\max}(\boldsymbol{x})]$. We can therefore define a function $\tau^\star$ mapping each $\boldsymbol{x}$ to the unique root of $h'(\tau, \boldsymbol{x})$ in $[\tau_{\min}(\boldsymbol{x}), \tau_{\max}(\boldsymbol{x})]$.

Step 2. We then show that $\tau^\star$ is continuous.

Note that, because $\mathcal{Y}$ is finite and $\boldsymbol{x} \mapsto q(\boldsymbol{y}|\boldsymbol{x})$, $\boldsymbol{x} \mapsto g(\boldsymbol{x}, \boldsymbol{y})$ and f' are continuous, both $\tau_{\min}$ and $\tau_{\max}$ are continuous functions of $\boldsymbol{x}$. Let $(\boldsymbol{x}_k)_k$ be a sequence such that $\boldsymbol{x}_k \to \boldsymbol{x}$ as $k \to +\infty$. Since $\tau_{\min}$ and $\tau_{\max}$ are continuous, $\tau_{\min}(\boldsymbol{x}_k) \to \tau_{\min}(\boldsymbol{x})$ and $\tau_{\max}(\boldsymbol{x}_k) \to \tau_{\max}(\boldsymbol{x})$. Therefore, $(\tau^\star(\boldsymbol{x}_k))_k$ is a bounded sequence. As such, it admits a converging subsequence $(\tau^\star(\boldsymbol{x}_{\varphi(k)}))_k$ converging to some $t^\star(\boldsymbol{x})$. One has $h'(\tau^\star(\boldsymbol{x}_{\varphi(k)}), \boldsymbol{x}_{\varphi(k)}) = 0$. By continuity of h', this gives $h'(t^\star(\boldsymbol{x}), \boldsymbol{x}) = 0$ and therefore $t^\star(\boldsymbol{x}) = \tau^\star(\boldsymbol{x})$ by unicity of the root. Therefore, any converging subsequence of the bounded sequence $(\tau^\star(\boldsymbol{x}_k))$ converges to $\tau^\star(\boldsymbol{x})$. This exactly proves that $\tau^\star(\boldsymbol{x}_k) \to \tau^\star(\boldsymbol{x})$ as $k \to +\infty$. Therefore, $\tau^\star$ is continuous and

$$\mathbb{E}_{\boldsymbol{x}} \min_{\tau_{\boldsymbol{x}} \in \mathbb{R}} \left[\tau_{\boldsymbol{x}} + \Omega^*_+(g_{\boldsymbol{x}} - \tau_{\boldsymbol{x}})\right] = \mathbb{E}_{\boldsymbol{x}} \left[\tau^\star(\boldsymbol{x}) + \Omega^*_+(g_{\boldsymbol{x}} - \tau^\star(\boldsymbol{x}))\right].$$

Step 3. Last, we show that $\min_{\tau \in \mathcal{F}(\mathcal{X})} \mathbb{E}_{\boldsymbol{x}} \left[\tau(\boldsymbol{x}) + \Omega^*_+(g_{\boldsymbol{x}} - \tau(\boldsymbol{x}))\right] = \mathbb{E}_{\boldsymbol{x}} \min_{\tau_{\boldsymbol{x}} \in \mathbb{R}} \left[\tau_{\boldsymbol{x}} + \Omega^*_+(g_{\boldsymbol{x}} - \tau_{\boldsymbol{x}})\right]$.

By definition of $\tau^\star$, we have $\min_{\tau \in \mathcal{F}(\mathcal{X})} \mathbb{E}_{\boldsymbol{x}} \left[\tau(\boldsymbol{x}) + \Omega^*_+(g_{\boldsymbol{x}} - \tau(\boldsymbol{x}))\right] \leq \mathbb{E}_{\boldsymbol{x}} \left[\tau^\star(\boldsymbol{x}) + \Omega^*_+(g_{\boldsymbol{x}} - \tau^\star(\boldsymbol{x}))\right] = \mathbb{E}_{\boldsymbol{x}} \min_{\tau_{\boldsymbol{x}} \in \mathbb{R}} \left[\tau_{\boldsymbol{x}} + \Omega^*_+(g_{\boldsymbol{x}} - \tau_{\boldsymbol{x}})\right]$. On the other hand, for any x and any $\tau \in \mathcal{F}(\mathcal{X})$, we have

$$\mathbb{E}_{\boldsymbol{x}} \left[\tau(\boldsymbol{x}) + \Omega^*_+(g_{\boldsymbol{x}} - \tau(\boldsymbol{x}))\right] \geq \mathbb{E}_{\boldsymbol{x}} \min_{\tau_{\boldsymbol{x}} \in \mathbb{R}} \left[\tau_{\boldsymbol{x}} + \Omega^*_+(g_{\boldsymbol{x}} - \tau_{\boldsymbol{x}})\right].$$

Since this holds for any continuous function τ, this gives

$$\min_{\tau \in \mathcal{F}(\mathcal{X})} \mathbb{E}_{\boldsymbol{x}} \left[\tau(\boldsymbol{x}) + \Omega^*_+(g_{\boldsymbol{x}} - \tau(\boldsymbol{x}))\right] \geq \mathbb{E}_{\boldsymbol{x}} \min_{\tau_{\boldsymbol{x}} \in \mathbb{R}} \left[\tau_{\boldsymbol{x}} + \Omega^*_+(g_{\boldsymbol{x}} - \tau_{\boldsymbol{x}})\right].$$

We then have

$$\min_{\tau \in \mathcal{F}(\mathcal{X})} \mathbb{E}_{\boldsymbol{x}} \left[\tau(\boldsymbol{x}) + \Omega^*_+(g_{\boldsymbol{x}} - \tau(\boldsymbol{x}))\right] = \mathbb{E}_{\boldsymbol{x}} \min_{\tau_{\boldsymbol{x}} \in \mathbb{R}} \left[\tau_{\boldsymbol{x}} + \Omega^*_+(g_{\boldsymbol{x}} - \tau_{\boldsymbol{x}})\right].$$

Therefore,

$$\min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \mathbb{E}_{\boldsymbol{x}} \left[\Omega^*_1(g_{\boldsymbol{x}}; q_{\boldsymbol{x}}) - \langle g_{\boldsymbol{x}}, \rho_{\boldsymbol{x}} \rangle\right] = \min_{g \in \mathcal{F}(\mathcal{X} \times \mathcal{Y})} \min_{\tau \in \mathcal{F}(\mathcal{X})} \mathbb{E}_{\boldsymbol{x}} \left[\tau(\boldsymbol{x}) + \Omega^*_+(g_{\boldsymbol{x}} - \tau(\boldsymbol{x})) - \langle g_{\boldsymbol{x}}, \rho_{\boldsymbol{x}} \rangle\right].$$

In addition, for all $(\boldsymbol{x}, \boldsymbol{y}) \in (\mathcal{X} \times \mathcal{Y})$

$$\pi_{g^\star, q}(\boldsymbol{y}|\boldsymbol{x}) = \nabla \Omega_+^*(g_{\boldsymbol{x}}^\star - \tau^\star(\boldsymbol{x}); q_{\boldsymbol{x}})(\boldsymbol{y}) = q(\boldsymbol{y}|\boldsymbol{x})(f_+^*)'(g^\star(\boldsymbol{x}, \boldsymbol{y})).$$

This concludes the proof.

C.6. Joint convexity

Proposition 5. *Given $q \in \mathcal{P}_+(\mathcal{Y}|\mathcal{X})$, f differentiable, strictly convex well defined on $(0, +\infty)$, for a coupling $g_{\boldsymbol{w}}(x, y)$ linear in $\boldsymbol{w} \in \mathcal{W}$ and a finite dimensional representation of the log-partition function, $\tau_{\boldsymbol{v}}(\boldsymbol{x}_i) = v_i$ for $i \in [n]$, the objective*

$$\widetilde{\mathcal{L}}_f(g_{\boldsymbol{w}}, \tau_{\boldsymbol{v}}) \coloneqq \frac{1}{n}\left(\sum_{i=1}^n L_{g_{\boldsymbol{w}}, \tau_{\boldsymbol{v}}}^f(\boldsymbol{x}_i) - g_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}_i)\right),$$
$$\text{where} \quad L_{g_{\boldsymbol{w}}, \tau_{\boldsymbol{v}}}^f(\boldsymbol{x}_i) \coloneqq \tau_{\boldsymbol{v}}(\boldsymbol{x}_i) + \sum_{\boldsymbol{y}' \in \mathcal{Y}} q(\boldsymbol{y}'|\boldsymbol{x}_i) f_+^*(g_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}') - \tau_{\boldsymbol{v}}(\boldsymbol{x}_i))$$

is jointly convex in $\boldsymbol{w}, \boldsymbol{v}$, where f_+^* is the convex conjugate of the restriction of f to $\mathbb{R}_+$.

Proof. The function $(\boldsymbol{w}, \boldsymbol{v}) \to g_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}') - \tau_{\boldsymbol{v}}(\boldsymbol{x}_i)$ is linear in $\boldsymbol{w}, \boldsymbol{v}$ by assumption. Since f_+^* is convex, $(\boldsymbol{w}, \boldsymbol{v}) \mapsto f_+^*(g_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}') - \tau_{\boldsymbol{v}}(\boldsymbol{x}_i))$ is then jointly convex in $\boldsymbol{w}, \boldsymbol{v}$ since the composition of a linear function and a convex function is always convex. Since $q(\boldsymbol{y}'|\boldsymbol{x}_i)$ is positive, $(\boldsymbol{w}, \boldsymbol{v}) \mapsto q(\boldsymbol{y}'|\boldsymbol{x}_i) f_+^*(g_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}') - \tau_{\boldsymbol{v}}(\boldsymbol{x}_i))$ is also jointly convex. The functions $\boldsymbol{w} \to g_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}_i)$ and $\boldsymbol{v} \mapsto \tau_{\boldsymbol{v}}(\boldsymbol{x}_i)$ are linear in $\boldsymbol{w}$ and $\boldsymbol{v}$ respectively. A sum of convex functions is convex hence the result. $\square$

C.7. Proof of gradient computation

One has

$$\begin{aligned}
\nabla_{\boldsymbol{w}} \mathrm{LSE}_{g_{\boldsymbol{w}}}(\boldsymbol{x}) &= \nabla_{\boldsymbol{w}} \log \sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) \exp(g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y})) \\
&= \frac{\sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) \nabla_{\boldsymbol{w}} \exp(g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}))}{\sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) \exp(g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}))} \\
&= \frac{\sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) \exp(g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y})) \nabla_{\boldsymbol{w}} g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y})}{\sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}|\boldsymbol{x}) \exp(g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}))} = \frac{\mathbb{E}_{\boldsymbol{y} \sim q(\cdot|\boldsymbol{x})}\left[\exp(g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y})) \nabla_{\boldsymbol{w}} g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y})\right]}{\mathbb{E}_{\boldsymbol{y} \sim q(\cdot|\boldsymbol{x})}\left[\exp(g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}))\right]} \\
&= \sum_{\boldsymbol{y} \in \mathcal{Y}} p_{g_{\boldsymbol{w}}}(\boldsymbol{y}|\boldsymbol{x}) \nabla_{\boldsymbol{w}} g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{\boldsymbol{y} \sim p_{g_{\boldsymbol{w}}}(\cdot|\boldsymbol{x})}\left[\nabla_{\boldsymbol{w}} g_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y})\right].
\end{aligned}$$