Relative Entropic Optimal Transport: a (Prior-aware) Matching Perspective to (Unbalanced) Classification

Liangliang Shi, Haoyu Zhen, Gu Zhang, Junchi Yan*
Dept. of Computer Science and Engineering & MoE Key Lab of AI, Shanghai Jiao Tong University
{shiliangliang, anye_zhen, blake-nash, yanjunchi}@sjtu.edu.cn
PyTorch Code: https://github.com/open-mmlab/mmrotate

Motivation: The inconsistency of the prior information (conditional distribution) between training and testing data.

Method: Rethinks classification from a matching perspective by studying the matching probability between samples and labels with optimal transport (OT) formulation.

1. Prior information and optimal transport (OT)
2. Relative entropic OT (RE-OT) with the specific smoothing prior Q.
3. (Long-tailed) Inverse RE-OT and epoch-varying smoothing-guided Q.
4. A significant link between Optimal Transport (OT) and classification tasks.
5. Image classification, molecule property prediction, and instance segmentation.

Consider two histograms $\mathbf{a} \in \Sigma_n$ and $\mathbf{b} \in \Sigma_m$, where the simplex $\Sigma_d = \{x \in \mathbb{R}_+^d | x^\top 1_d = 1\}$. We can represent the transportation polytope $U(\mathbf{a}, \mathbf{b})$, which is the polyhedral set of $n \times m$ matrices:

$$U(\mathbf{a}, \mathbf{b}) = \{\mathbf{P} \in \mathbb{R}_+^{n \times m} | \mathbf{P}\mathbf{1}_m = \mathbf{a}, \mathbf{P}^\top \mathbf{1}_n = \mathbf{b}\}, \tag{1}$$

where $\mathbf{1}_n$ and $\mathbf{1}_m$ are n and m dimensional vectors of ones. $U(\mathbf{a}, \mathbf{b})$ contains all $n \times m$ nonnegative probabilistic matrices with row and column sums $\mathbf{a}$ and $\mathbf{b}$. Then the Kantorovich's optimal transport problem can be defined as [22]:

$$\min_{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})} < \mathbf{C}, \mathbf{P} >= \sum_{i,j} \mathbf{C}_{ij} \mathbf{P}_{ij}, \tag{2}$$

where $\mathbf{C} \in \mathbb{R}_+^{n \times m}$ is the cost matrix for the distance between sample $\{\mathbf{x}_i\}_{i=1}^n$ and $\{\mathbf{y}_j\}_{j=1}^m$. The above minimization is a linear program and the optimal solution $\mathbf{P}^*$ can be obtained with the network simplex [46] or other off-the-shelf techniques e.g. [35], which require significant time overhead. For its cost-effectiveness, the entropic regularization [49] is more popular and it gets an approximation:

$$\min_{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})} \mathcal{L}^\epsilon =< \mathbf{C}, \mathbf{P} > -\epsilon H(\mathbf{P}), \tag{3}$$

where the regularizer is specified as $H(\mathbf{P}) = \sum_{ij} -\mathbf{P}_{ij}(\log(\mathbf{P}_{ij}) - 1)$. It can be proved that [38] when $\epsilon \to 0$, the unique solution of Eq. 3 converges to the optimal $\mathbf{P}^*$ of original problem. And when $\epsilon \to +\infty$, we can get

$$\mathbf{P}_\epsilon \overset{\epsilon \to +\infty}{\longrightarrow} \mathbf{a} \otimes \mathbf{b}, \tag{4}$$

where $\mathbf{a} \otimes \mathbf{b} = \mathbf{a}^\top \mathbf{b} \in U(\mathbf{a}, \mathbf{b})$. So we can find the entropic regularization $H(\mathbf{P})$ exactly do a smoothing to the solution towards $\mathbf{a} \otimes \mathbf{b}$. Recently, Inverse Optimal Transport (IOT) start to be studied [11, 29, 43] which assumes that the cost $\mathbf{C}$ is unknown for learning, given the established coupling. The inverse of entropic OT problem can be formulated as:

$$\min_\theta KL(\tilde{\mathbf{P}} | \mathbf{P}^\theta), \quad \text{where} \quad \mathbf{P}^\theta = \arg \min_{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})} < \mathbf{C}^\theta, \mathbf{P} > -\epsilon H(\mathbf{P}). \tag{5}$$

1. Prior information and optimal transport (OT)

Motivation of introducing prior into OT. As introduced in Eq. 3, entropic regularization (ER) is an effective method [10] for approximating the Kantorovich OT solution. It pushes the original linear programming solution away from the hard boundary and obtains a smoother solution by minimizing a $\epsilon-$strongly convex function. With the increase of the penalty coefficient, the solution progressively approaches a more 'smooth' solution, ultimately converging to $\mathbf{a}^\top \mathbf{b}$ as illustrated by Eq. 4. However, a question is raised: is such a smoothing direction $\mathbf{a}^\top \mathbf{b}$ appropriate for all practical situations, e.g. the long-tailed problems? Can we freely choose a more suitable (smoothing) direction to achieve more effective results tailored to the problem or dataset at hand? In this section, we will develop relative entropic OT (RE-OT), a more general formulation than traditional entropic OT [49].

2. Relative entropic OT (RE-OT) and specific smoothing prior Q

Formulation. Given a cost matrix $\mathbf{C} \in \mathbb{R}_{+}^{n \times m}$ and two margins $\mathbf{a} \in \Sigma_n$ and $\mathbf{b} \in \Sigma_m$, where Σ_n and Σ_m represent the probability simplex with n and m bins, respectively, we introduce the positive smoothing-guidance matrix $\mathbf{Q}$ as a manually-specified constant prior to guide the smoothing. We use the relative entropy regularizer $H_Q(\mathbf{P})$ in place of the traditional entropy in Eq. 3 to achieve this ability, defined as $H_Q(\mathbf{P}) = -\sum_{ij} \mathbf{P}_{ij} \left(\log \frac{\mathbf{P}_{ij}}{\mathbf{Q}_{ij}} - 1 \right)$. Then we have:

$$\min_{\mathbf{P} \in U(\mathbf{a},\mathbf{b})} \mathcal{L}_{\mathbf{Q}}^{\epsilon} = < \mathbf{C}, \mathbf{P} > -\epsilon H_Q(\mathbf{P}). \tag{6}$$

The relative entropy $H_Q(\mathbf{P})$ can increase the probability of transportation for the element $\mathbf{P}_{ij}$ if $\mathbf{Q}_{ij}$ is larger. Consequently, if the prior in the form of $\mathbf{Q}$ is available, the use of relative entropy regularization can improve the efficiency of learning the transportation solution. Therefore, for long-tailed tasks, we can assign a large $\mathbf{Q}$ element to the majority classes and a low element of $\mathbf{Q}$ to the tailed classes to enhance training. We will discuss this in detail in the next section.

Proposition 1 (**Solution Propriety.**). *Assume* $\mathbf{P}_Q^\epsilon, \mathbf{P}^\epsilon$ *and are the optimal solution of RE-OT and entropic regularized OT, respectively. Then we can get:*

(1) When $\epsilon \to +\infty$, *the optimal RE-OT's solution* $\mathbf{P}_Q^\epsilon$ *will converge to* $\widetilde{\mathbf{Q}}$ *where* $\widetilde{\mathbf{Q}}$ *takes the form* $\widetilde{\mathbf{Q}} = diag(\mathbf{u})\mathbf{Q}diag(\mathbf{v})$ *with two uniquely defined non-negative vectors* $\mathbf{u}$ *and* $\mathbf{v}$.

(2) With the prior $\mathbf{Q}$ *and its corresponding* $\widetilde{\mathbf{Q}}$ *as defined in (1), we have* $\mathbf{P}_{\mathbf{Q}}^\epsilon = \mathbf{P}_{\widetilde{\mathbf{Q}}}^\epsilon$. *And when* $\widetilde{\mathbf{Q}} = \mathbf{a} \otimes \mathbf{b}$, *we have the equality* $\mathbf{P}^\epsilon = \mathbf{P}_{\widetilde{\mathbf{Q}}}^\epsilon$.

Proposition 2 (**static Schrödinger form**). *Redefine a general KL divergence as*

$$\widetilde{KL}(\mathbf{P}|\mathbf{K}) = \sum_{ij} \mathbf{P}_{ij} \log \frac{\mathbf{P}_{ij}}{\mathbf{K}_{ij}} - \mathbf{P}_{ij} + \mathbf{K}_{ij}, \tag{7}$$

The optimization in Eq. 6 is equivalent to the following minimization, where $\mathbf{K}_{ij} = \mathbf{Q}_{ij} e^{-\mathbf{C}_{ij}/\epsilon}$:

$$\mathbf{P}_Q^\epsilon = \underset{\mathbf{P} \in U(\mathbf{a},\mathbf{b})}{\arg\min}\ \widetilde{KL}(\mathbf{P}|\mathbf{K}). \tag{8}$$

Proposition 3 (**Dual formulation**). *From the optimization in Eq. 6, we can get its dual formulation:*

$$L_{\mathbf{Q}}^\epsilon = < \mathbf{f}, \mathbf{a} > + < \mathbf{g}, \mathbf{b} > - \epsilon < e^{\mathbf{f}/\epsilon}, \mathbf{K} e^{\mathbf{g}/\epsilon} > \tag{11}$$

where $\mathbf{f} \in \mathbb{R}^n$ *and* $\mathbf{g} \in \mathbb{R}^m$ *are the corresponding dual variables.*

The proof is given in Appendix D. Exactly $\mathbf{f}, \mathbf{g}$ are linked to $\mathbf{u}, \mathbf{v}$ appearing in Sinkhorn algorithm by $\mathbf{u} = \exp(\mathbf{f}/\epsilon)$ and $\mathbf{v} = \exp(\mathbf{g}/\epsilon)$, and thus Sinkhorn algorithm can be done in log-domain [41].

The Sinkhorn algorithm and barycenters for RE-OT. The optimal solution for RE-OT can be estimated using the Sinkhorn algorithm, as its counterpart has already been well-developed for solving entropic OT [10]. The algorithm for RE-OT shares the same form as that for entropic OT, given $\mathbf{K}_{ij} = \widetilde{\mathbf{Q}}_{ij} e^{-\mathbf{C}_{ij}/\epsilon}$. Specifically, with two non-negative vectors $\mathbf{u}'$ and $\mathbf{v}'$ uniquely defined up to a multiplicative factor, the optimal solution has the form $\mathbf{P}^{\epsilon}_{Q} = \mathrm{diag}(\mathbf{u}')\mathbf{K}\,\mathrm{diag}(\mathbf{v}')$, which can be efficiently computed by iterating $\mathbf{u}', \mathbf{v}' \leftarrow \mathbf{a}./K\mathbf{v}', \mathbf{b}./\mathbf{K}^{\top}\mathbf{u}'$. The proof and algorithm are presented in Appendix C. In addition, the barycenter between distributions is a natural extension of OT [1, 2], and with the matrix $\mathbf{K}$, [2] defined it with a weighted KL projection problem:

$$\min_{\{\mathbf{P}_s\},\mathbf{a}} \sum_s \epsilon \cdot \lambda_s \widetilde{KL}(\mathbf{P}_s|\mathbf{K}) \text{ s.t. } \mathbf{P}_s^{\top}\mathbf{1}_n = \mathbf{b}_s, \mathbf{P}_1\mathbf{1}_m = \mathbf{P}_2\mathbf{1}_m = \cdots = \mathbf{P}_S\mathbf{1}_m = \mathbf{a}, \tag{9}$$

where $\{\lambda_s\}$ are the weights, $\mathbf{b}_s$ are known histograms representing images, and $\mathbf{a}$ is the barycenter histogram to be calculated. Though easier to calculate compared with the regularized formulation in [1], a drawback of this entropic-based method is that it can lead to blurred barycenters, and methods have been proposed to address this issue [19]. By calculating barycenters between noise and an image, we show that the blurred problem can be simply solved by setting $\mathbf{Q}$:

$$\mathbf{Q} = (1-\lambda)(\mathbf{P}^{\epsilon})^{\top} + \lambda\mathbf{P}^{\epsilon}, \tag{10}$$

where $\mathbf{P}^{\epsilon}$ is the optimal solution of entropic OT from image $\mathbf{b}_1$ to $\mathbf{b}_2$. Fig. 2 illustrates the effect of different choices of $\mathbf{Q}$ when computing barycenters between noise and an image. Without using $\mathbf{Q}$, the resulting barycenters are very blurry. When we set $\mathbf{Q}$ to be $\mathbf{P}^{\epsilon}$, the barycenters display the shape of the leopard clearly, but fail to transition smoothly from noise to leopard image as ϵ changes. To address this issue, we set $\mathbf{Q}$ as shown in Eq. 10, which can blur the image while transitioning smoothly with the gradient.

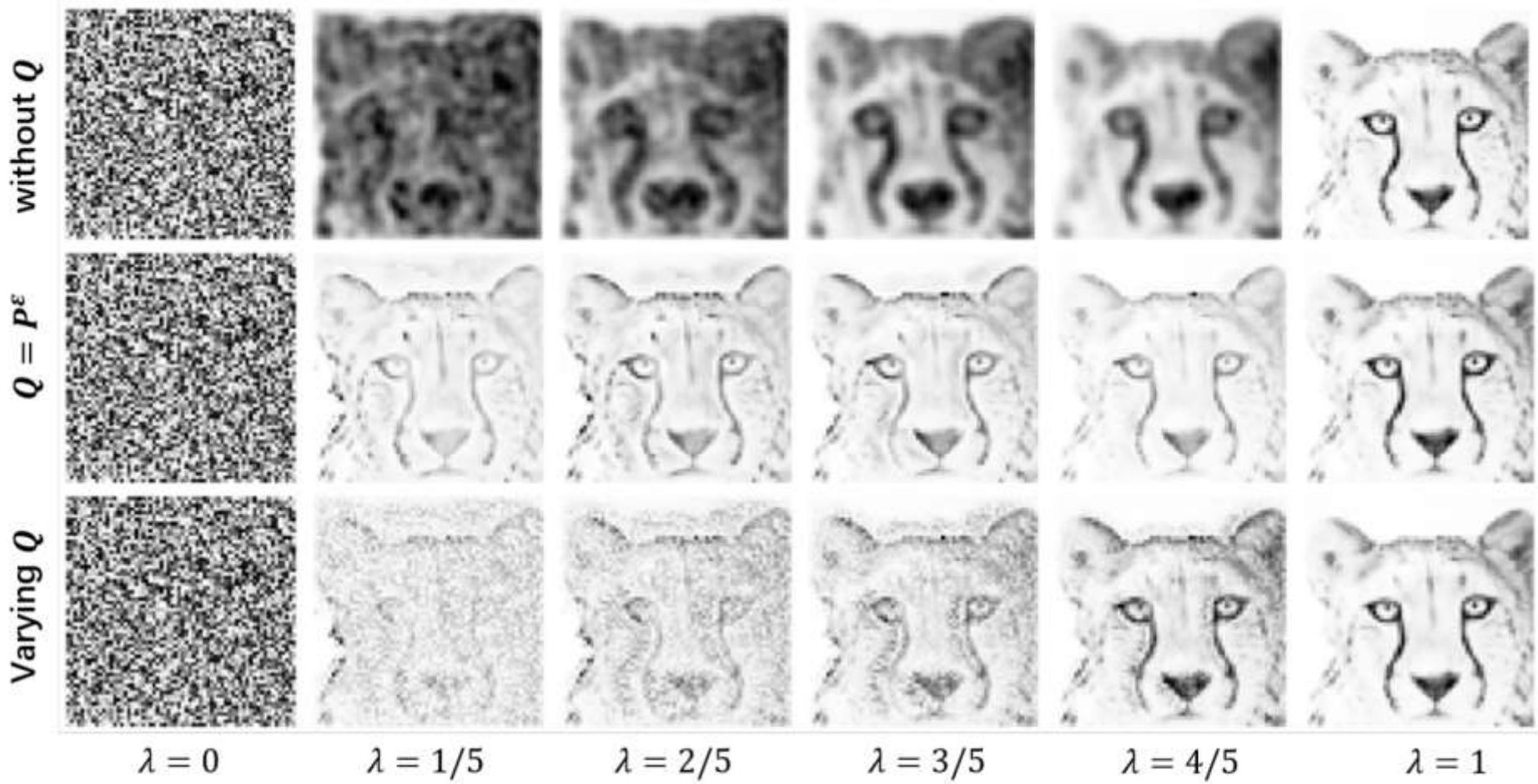

Figure 2: The barycenter results between noise and a leopard image. It is very blurry without the use of $\mathbf{Q}$. However, when we set $\mathbf{Q} = \mathbf{P}^{\epsilon}$, the resulting barycenters clearly display the shape of the leopard, but fail to transition smoothly from noise to leopard image as λ changes. To address this, we set $\mathbf{Q}$ as shown in Eq. 10, which blurs the image while transitioning with the increase of λ.

Setting Q with the Optimal Solution Iteratively. The intuition for the prior is to iteratively update the solution $P^{\epsilon}_{\mathbf{Q}}$ as a new $\mathbf{Q}$, i.e., $\mathbf{Q}^{(n)} = \mathbf{P}^{\epsilon}_{\mathbf{Q}^{(n-1)}}$, where $\mathbf{Q}^{(n)}$ represents the solution obtained after the n-th iteration. The question then arises as to how $\mathbf{Q}^{(n)}$ changes over time. We find that this problem is equivalent to a proximal point algorithm:

$$\mathbf{Q}^{(n)} = \arg\min_{\mathbf{Q}} KL(\mathbf{Q}|\mathbf{Q}^{(n-1)}) + \frac{1}{\epsilon}F(\mathbf{Q}). \tag{12}$$

Here $F(\mathbf{Q}) =< C, \mathbf{Q} > + l_{U(\mathbf{a},\mathbf{b})}(\mathbf{Q})$, where $l_{U(\mathbf{a},\mathbf{b})}(\mathbf{Q})$ is defined such that $l_{U(\mathbf{a},\mathbf{b})}(\mathbf{Q}) = 0$ if $\mathbf{Q} \in U(\mathbf{a}, \mathbf{b})$, and $l_{U(\mathbf{a},\mathbf{b})}(\mathbf{Q}) = +\infty$ otherwise. We find this problem is discussed by [25, 38],

which aims to get the optimal solution of the non-regularized OT. As discussed in Sinkhorn of RE-OT, we have the optimal solution form:

$$\begin{aligned}\mathbf{Q}^{(n)} &= \mathrm{diag}(\mathbf{u}^{(n-1)})\mathbf{Q}^{(n-1)} \odot e^{-C/\epsilon}\,\mathrm{diag}(\mathbf{v}^{(n-1)}) \\ &= \mathrm{diag}(\mathbf{u}^{(n-1)} \odot ... \odot \mathbf{u}^{(0)})\mathbf{Q}^{(0)} \odot e^{-\frac{(n+1)C}{\epsilon}}\,\mathrm{diag}(\mathbf{v}^{(0)} \odot ...\mathbf{v}^{(n-1)}).\end{aligned} \tag{13}$$

When we set $\mathbf{Q}^{(0)} = \mathbf{1}_{n\times m}$, the optimization is equivalent to applying Sinkhorn's algorithm iteratively with a kernel $e^{-\frac{(n+1)C}{\epsilon}}$, i.e., with a decaying regularization coefficient $\frac{\epsilon}{n+1}$. This observation further emphasizes the importance of selecting an appropriate $\mathbf{Q}$ for RE-OT.

3. (Long-tailed) Inverse RE-OT and epoch-varying smoothing-guided Q

We now discuss the application of inverse RE-OT to address the long-tailed classification task (including the contrastive learning setting). Unlike previous works that study this problem using conditional probability, our approach aims to learn features by matching samples with their labels using Inverse OT. Specifically, given the mini-batch pair set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ where $\mathbf{x}_i$ is training sample and $\mathbf{y}_i$ is the $m-$dimensional one-hot label of $\mathbf{x}_i$. We can set the the cost $\mathbf{C}_{ij}$ between sample $\mathbf{x}_i$ and another one-hot label given the neural network f_θ. Without loss of generality, we can set $\mathbf{C}_{ij} = c - l_{ij}$ where $l_{ij} = (f_\theta(\mathbf{x}_i))_j$ is the $j-$th component of logits $f_\theta(\mathbf{x}_i)$ for sample $\mathbf{x}_i$. Then we can learn the matching between features and labels with inverse OT via a bi-level optimization:

$$\min_\theta KL(\widetilde{\mathbf{P}}|\mathbf{P}^\theta) \quad s.t. \quad \mathbf{P}^\theta = \arg\min_{\mathbf{P}\in U} < \mathbf{C}, \mathbf{P} > -\epsilon H_Q(\mathbf{P}), \tag{14}$$

where $\widetilde{\mathbf{P}}$ is the supervision given the label $\{\mathbf{y}_i\}$. For example, we can set $\widetilde{\mathbf{P}}_{ij} = \mathbf{y}_{ij}$ when $\mathbf{y}_{ij}$ is the j-th component of $\mathbf{y}_i$. Here U refer to the constraints for the coupling $\mathbf{P}$. We can set $U = U(\mathbf{a}, \mathbf{b})$ with the full matching constraints or its relaxation e.g. $U = U(\mathbf{a}) = \{\mathbf{P}|\mathbf{P}\mathbf{1}_m = \mathbf{a}\}$.

We set $\mathbf{Q}$ by varying over epochs:

$$\mathbf{Q} = (1 - \lambda(t))\,\text{Uniform} + \lambda(t)\mathbf{r}, \tag{33}$$

where we let $\lambda(t)$ be a piece-wise linear function:

$$\lambda(t) = \begin{cases} 0 & \text{if } t < t_1 \\ \frac{t-t_1}{t_2-t} & \text{if } t_1 < t < t_2 \\ 1 & \text{if } t_2 < t < T \end{cases}. \tag{34}$$

Here t is the training epoch number and t_1, t_2, t_3 are hyper-parameters. And for the setting of $\mathbf{r}$, a simple set is the balanced ratio as used in [39] which is specified as

$$\mathbf{r}_{ij} = \frac{n_j}{\sum_k n_k}, \tag{35}$$

where n_j is the number of samples in j class. Or we can set $\mathbf{r}$ as

$$\mathbf{r}_{ij} = \frac{n_j^b}{n} \tag{36}$$

where n_j^b is the mini-bath samples' number in j class and n is the batchsize. We can also adopt the interpolation of Eq. 35 and Eq. 36, which is specified as

$$\mathbf{r} = \gamma \times \frac{n_j}{\sum_k n_k} + (1 - \gamma) \times \frac{n_j^b}{n}, \tag{37}$$

where γ is a smoothing parameter determining the degree of blending between Eq. 36 and Eq. 35. The above three are all class-wise and we can also define $\mathbf{r}$ with class-sample-wise form with

$$\mathbf{r}_{ij} = \frac{e^{l_i j}}{\sum_k e^{l_i k}} * \frac{n_j}{\sum_k n_k}, \tag{38}$$

where $\frac{e^{l_i j}}{\sum_k e^{l_i k}}$ is used to control the simple-wise factor with different sample confidences.

$$\min_{\theta} KL(\widetilde{\mathbf{P}}|\mathbf{P}^{\theta}) \quad s.t. \quad \mathbf{P}^{\theta} = \arg\min_{\mathbf{P}\in U} < \mathbf{C}, \mathbf{P} > -\epsilon H_Q(\mathbf{P}), \tag{14}$$

$$H_Q(\mathbf{P}) = -\sum_{ij} \mathbf{P}_{ij} \left(\log \frac{\mathbf{P}_{ij}}{\mathbf{Q}_{ij}} - 1 \right)$$

or the sample-class-wise setting discussed in Appendix E. $\lambda(t)$ is a epoch varying weight. When t is small, $\lambda(t)$ is close to 0, and $\mathbf{Q}$ is more likely to be a uniform distribution, which gives the model less prior information. As t approaches the final training epoch number T, $\lambda(t) \to 1$ and $\mathbf{Q} \to \mathbf{r}$, which means the model is given the full prior information. This gradual process of introducing prior information can be helpful for training the inverse OT. We provide a specific setting for $\lambda(t)$ and $\mathbf{r}$ in Appendix E. During the inference process, the problem reduces to the inner optimization in Eq. 6, which takes the form of traditional unsupervised OT. We use this optimization to obtain the matching between the sample features and labels, which corresponds to classification for the testing data. We do not adopt the long-tailed setting of $\mathbf{Q}$ during inference because the testing data is assumed to be uniform.

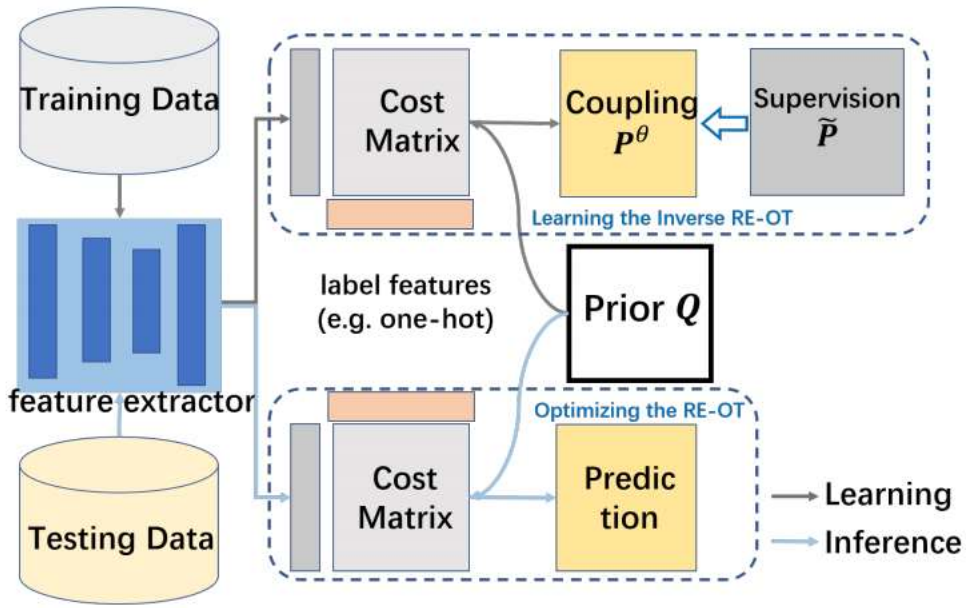

Figure 3: RE-OT for unbalanced classification.

4. A significant link between Optimal Transport (OT) and classification tasks.

The following table is a more complete results in Tab. 1, where $\gamma, \beta, \beta_{y_i}, n_j$ are all hyper-parameters and ψ is the cosine-based function to learn the feature in hyper-sphere space in A-softmax. And for triplet loss [42], $f(\cdot)$ is the feature extractor to learn the representations. More details in Appendix H for triplet loss.

Table 5: Previous works in the view of inverse RE-OT by setting different ground truth $\tilde{\mathbf{P}}_{ij}$, cost $\mathbf{C}_{ij}$, prior $\mathbf{Q}_{ij}$ and coefficient ϵ under $U(\mathbf{a})$. More details are given in Appendix G.

Methods	**Formulation**			
	Ground Truth $\tilde{\mathbf{P}}_{ij}$	Cost $\mathbf{C}_{ij}$	Prior $\mathbf{Q}_{ij}$	penalty coefficient ϵ
Vanilla Softmax	$\tilde{\mathbf{P}}_{ij} = y_{ij}$	$\mathbf{C}_{ij} = c - l_{ij}$	$Q_{ij} = 1$	$\epsilon = \tau$
Focal Loss [30]	$\tilde{\mathbf{P}}_{ij} = y_{ij} * (1 - \mathbf{P}_{ij})^{\gamma}$	$\mathbf{C}_{ij} = c - l_{ij}$	$Q_{ij} = 1$	$\epsilon = \tau$
Balanced-Softmax [39]	$\tilde{\mathbf{P}}_{ij} = y_{ij}$	$\mathbf{C}_{ij} = c - l_{ij}$	$Q_{ij} = n_j$	$\epsilon = \tau$
Class-Balanced Loss [8]	$\tilde{\mathbf{P}}_{ij} = (1-\beta)/\left(1-\beta^{n}_{y_{i,:}}\right)$	$\mathbf{C}_{ij} = c - l_{ij}$	$Q_{ij} = 1$	$\epsilon = \tau$
LDAM Loss [3]	$\tilde{\mathbf{P}}_{ij} =$	$\mathbf{C}_{ij} = c - l_{ij}$	$Q_{ij} = e^{-\mathbf{1}\{i=y\} \cdot C/n_j^{\gamma}}$	$\epsilon = \tau$
A-softmax [32]	$\tilde{\mathbf{P}}_{ij} = y_{ij}$	$\mathbf{C}_{ij} = c - \|l_i\| \psi(\theta_{\mathbf{y}_i,i})$	$Q_{ij} = 1$	$\epsilon = \tau$
Triplet [42]	$\tilde{\mathbf{P}}_{ij} = y_{ij}$	$\mathbf{C}_{ij} = \|\|f(x_i) - f(x_j)\|\|^2$	$Q_{ij} = 1$	$\epsilon \to 0^+$

Definition 1 (Barycentric Projection). *Consider the setting of Eq. 6 in which we use entropic regularization to approximate OT between discrete measures. One can define the so-called barycentric projection map with the coupling* $\mathbf{P}$

$$T : x_i \in X \rightarrow \frac{1}{\mathbf{a}_i} \sum\nolimits_j \mathbf{P}_{ij} y_j \in Y, \tag{17}$$

where $\{x_i\} \subset X$ *is the set of locations croponding to simplex* $\mathbf{a}$ *and* $\{y_j\}$ *is the set of locations in the* Y *space croponding to simplex* $\mathbf{b}$.

From the classification view, exactly $T(x_i)$ is probability confidences of samples x_i which can be understand as the feature in the one-hot label space. Besides, motivated by this barycentric projection in OT, we can exactly define the mapping for the feature of one-hot label in hidden feature space:

$$T' : y_j \in Y \rightarrow \frac{1}{\sum_i \mathbf{P}_{ij}} \sum\nolimits_i \mathbf{P}_{ij} \mathbf{f}_i \in \mathcal{F}, \tag{18}$$

where $\mathbf{f}_i$ is the feature of x_i extracting from the neural networks and $\mathcal{F}$ is the feature space. Eq. 18 maps the (one-hot) label to the feature space, which means no need to learn the centroid features as done in oneline clustering methods [4, 28] but calculate it statically with Eq. 18 if the coupling is known. Fig. 4 shows the results of t-SNE for the sample logit features and the class barycenters.

5. Image classification, molecule property prediction, and instance segmentation.

Table 3: Comparison on CIFAR10/100-LT with constrastive methods.Top-1 accuracy (%) is reported with 100 imbalanced ratio.

Method	CIFAR10-LT		CIFAR100-LT	
	Lin.	KNN	Lin.	KNN
SimCLR [5]	56.70	46.99	29.27	21.90
SupCon [23]	72.60	70.50	**41.16**	38.14
Ours	**73.51**	**72.51**	40.92	**38.30**

Table 4: Top-1 accuracy (%) for long-tailed image classification with 200 imbalanced factor.

Method	CIFAR10-LT			CIFAR100-LT				ImageNet-LT			
	Many	Few	All	Many	Medium	Few	All	Many	Medium	Few	All
Vanilla Softmax	77.4	68.9	74.9	75.8	48.2	11.0	42.0	57.3	26.2	3.1	35.0
Focal Loss [30]	79.6	58.4	73.3	**76.1**	46.9	11.1	41.7	57.3	27.6	4.4	35.9
LDAM [3]	80.5	65.2	75.9	75.7	50.6	11.5	42.9	**57.3**	27.6	4.4	35.9
LogitAdjust [34]	80.0	35.3	66.6	75.7	39.2	4.1	36.5	54.2	14.0	0.4	27.6
CB-CE [8]	76.6	70.7	74.8	53.2	48.8	13.3	36.3	35.3	32.1	21.2	31.9
CB-FC [8]	76.6	70.7	74.8	53.2	48.8	13.3	36.3	35.3	32.1	**21.2**	31.9
Balanced Softmax [39]	**82.2**	71.6	79.0	70.3	50.4	26.5	47.0	52.5	38.6	17.8	41.1
Ours	81.5	**74.6**	**79.4**	70.4	**53.0**	**26.6**	**47.9**	53.5	**39.0**	17.4	**41.6**

Table 2: Test on instance segmentation and molecule classification.

Method	**LVIS (instance seg.)**				**OGBG-MOLBBBP**	**OGBG-MOLBACE**
	AP_m	AP_r	AP_c	AP_f	ROC-AUC	ROC-AUC
Vanilla	20.62	9.66	19.11	27.11	69.21 ± 0.34	79.40 ± 1.20
Focal Loss [30]	19.69	8.75	17.74	26.68	68.15 ± 1.29	80.65 ± 1.50
LDAM [3]	15.53	7.14	13.23	20.89	65.99 ± 0.94	79.20 ± 2.00
LogitAdjust [34]	22.41	11.70	21.18	**28.48**	69.05 ± 1.59	81.25 ± 0.33
CB-CE [8]	20.38	9.58	18.72	26.98	69.19 ± 1.18	79.39 ± 1.16
CB-FC [8]	21.41	**15.67**	20.34	25.25	69.51 ± 1.18	80.24 ± 1.45
Balanced Softmax	22.60	12.88	21.20	28.44	68.13 ± 0.87	80.26 ± 2.28
Ours	**22.64**	13.15	**21.26**	28.34	**70.48 ± 0.73**	**82.48 ± 1.59**