WASSERSTEIN BARYCENTER MODEL ENSEMBLING

Pierre Dognin∗, Igor Melnyk∗, Youssef Mroueh∗, Jerret Ross∗, Cicero Dos Santos∗ & Tom Sercu∗
IBM Research & MIT-IBM Watson AI Lab
∗ Alphabetical order; Equal contribution
{pdognin,mroueh,rossja,cicerons}@us.ibm.com,
{igor.melnyk,tom.sercu1}@ibm.com

ABSTRACT

In this paper we propose to perform model ensembling in a multiclass or a multilabel learning setting using Wasserstein (W.) barycenters. Optimal transport metrics, such as the Wasserstein distance, allow incorporating semantic side information such as word embeddings. Using W. barycenters to find the consensus between models allows us to balance confidence and semantics in finding the agreement between the models. We show applications of Wasserstein ensembling in attribute-based classification, multilabel learning and image captioning generation. These results show that the W. ensembling is a viable alternative to the basic geometric or arithmetic mean ensembling.

1 INTRODUCTION

Model ensembling consists in combining many models into a stronger, more robust and more accurate model. Ensembling is ubiquitous in machine learning and yields improved accuracies across multiple prediction tasks such as multi-class or multi-label classification. For instance in deep learning, output layers of Deep Neural Networks (DNNs), such as softmaxes or sigmoids, are usually combined using a simple arithmetic or geometric mean. The arithmetic mean rewards confidence of the models while the geometric means seeks the consensus across models.

What is missing in the current approaches to models ensembling, is the ability to incorporate side information such as class relationships represented by a graph or via an embedding space. For example a semantic class can be represented with a finite dimensional vector in a pretrained word embedding space such as GloVe (Pennington et al., 2014). The models’ predictions can be seen as defining a distribution in this label space defined by word embeddings: if we denote \( p_i \) to be the confidence of a model on a bin corresponding to a word having an embedding \( x_i \), the distribution on the label space is therefore \( p = \sum_i p_i \delta_{x_i} \). In order to find the consensus between many models predictions, we propose to achieve this consensus within this representation in the label space. In contrast to arithmetic and geometric averaging, which are limited to the independent bins’ confidence, this has the advantage of carrying the semantics to model averaging via the word embeddings. More generally this semantic information can be encoded via cost a matrix \( C \), where \( C_{ij} \) encodes the dissimilarity between semantic classes \( i \) and \( j \), and \( C \) defines a ground metric on the label space.

To achieve this goal, we propose to combine model predictions via Wasserstein (W.) barycenters (Agueh & Carlier, 2011), which enables us to balance the confidence of the models and the semantic side information in finding a consensus between the models. Wasserstein distances are a naturally good fit for such a task, since they are defined with respect to a ground metric in the label space of the models, which carry such semantic information. Moreover they enable the possibility of ensembling predictions defined on different label sets, since the Wasserstein distance allows to align and compare those different predictions. Since their introduction in (Agueh & Carlier, 2011) W. barycenter computations were facilitated by entropic regularization (Cuturi, 2013) and iterative algorithms that rely on iterative Bregman projections (Benamou et al., 2015). Many applications have used W. barycenters in Natural Language Processing (NLP), clustering and graphics. We show in this paper that W. barycenters are effective in model ensembling and in finding a semantic consensus, and can be applied to a wide range of problems in machine learning (Table 1).

The paper is organized as follows: In Section 2 we revisit geometric and arithmetic means from a geometric viewpoint, showing that they are \( \ell_2 \) and Kullback Leibler divergence KL (extended
2 \textbf{WASSERSTEIN BARYCENTERS FOR MODEL ENSEMBLING}

\textbf{Normalized and Unnormalized predictions Ensembling.} In deep learning, predictions on a label space of fixed size $M$ are usually in one of two forms: a) normalized probabilities: in a multi-class setting, the neural network outputs a probability vector (normalized through softmax), where each bin corresponds to a semantic class; b) unnormalized positive scores: in a multi-label setting, the outputs of $M$ independent logistic units are unnormalized positive scores, where each unit corresponds to the presence or the absence of a semantic class.

Model ensembling in those two scenarios has long history in deep learning and more generally in machine learning (Breiman, 1996; Freund & Schapire, 1999; Wolpert, 1992) as they lead to more robust and accurate models. As discussed in the introduction, two methods have been prominent in model ensembling due to their simplicity: majority vote using the arithmetic mean of predictions, or consensus based using the geometric mean.

\textbf{Revisiting Arithmetic and Geometric Means from a geometric viewpoint.} Given $m$ predictions $\mu_\ell$, and weights $\lambda_\ell \geq 0$ such that $\sum_{\ell=1}^{m} \lambda_\ell = 1$, the weighted arithmetic mean is given by $\bar{\mu}_a = \sum_{\ell=1}^{m} \lambda_\ell \mu_\ell$, and the weighted geometric mean by $\bar{\mu}_g = \prod_{\ell=1}^{m} (\mu_\ell^{\lambda_\ell})$.

It is instructive to reinterpret the arithmetic and geometric mean as weighted Frechet means (Definition 1) (Zemel & Panaretos, 2017).

\textbf{Definition 1.} [Weighted Frechet Mean] Given a distance $d$ and $\{(\lambda_\ell, \mu_\ell), \lambda_\ell > 0, \mu_\ell \in \mathbb{R}_+^M\}_{\ell=1}^m$, the Frechet mean is defined as follows: $\bar{\mu} = \arg \min_{\rho} \sum_{\ell=1}^{m} \lambda_\ell d(\rho, \mu_\ell)$.

It is easy to prove (Appendix F) that the arithmetic mean corresponds to a Frechet mean for $d = \| \cdot \|_2^2$ (the $\ell_2$ Euclidean distance). A less known fact is that the geometric mean corresponds to a Frechet Mean for $d = \tilde{K}_L$, where $\tilde{K}_L$ is the extended KL divergence to unnormalized measures: $\tilde{K}_L(p, q) = \sum_i p_i \log \left( \frac{p_i}{q_i} \right) - p_i + q_i$. We give proofs and properties of arithmetic and geometric mean in Appendix F.

Following this geometric viewpoint, in order to incorporate the semantics of the target space in model ensembling, we need to use a distance $d$ that takes advantage of the underlying geometry of the label space via a cost matrix $C$ when comparing positive measures. Optimal transport (OT) metrics such as Wasserstein-2 have this property since they are built on an explicit cost matrix defining pairwise distance between the semantic classes. In this paper we propose to use the Frechet means with Wasserstein distance $(d = W_2^2)$ for model ensembling, i.e. use Wasserstein barycenters (Agueh & Carlier, 2011) for model ensembling:

$$\bar{\mu}_w = \arg \min_{\rho} \sum_{\ell=1}^{m} \lambda_\ell W_2^2(\rho, \mu_\ell).$$

Intuitively, the barycenter looks for a distribution $\rho$ (a histogram) that is close to all the base distributions $\mu_\ell$ in the Wasserstein sense. In our context transporting the consensus $\rho$ to each individual model $\mu_\ell$ should have a minimal cost, where the cost is defined by the distance in the word embedding space.

3 \textbf{WASSERSTEIN BARYCENTERS}

Wasserstein distances were originally defined between normalized probability vectors (Balanced OT) (Villani, 2008; Peyré & Cuturi, 2017), but they have been extended to deal with unnormalized measures and this problem is referred to as unbalanced OT (Chizat et al., 2018; Frohner et al., 2015). Motivated by the multi-class and the multi-label ensembling applications, in the following we present a brief overview of W. barycenters in the balanced and unbalanced cases.
3.1 Optimal Transport Metrics

Balanced OT. Given \( p \in \Delta_N \), where \( \Delta_N = \{ p \in \mathbb{R}^N, p_k \geq 0, \sum_{k=1}^{N} p_k = 1 \} \), \( p \) represents histograms on source label space \( \Omega^S = \{ x_i \in \mathbb{R}^d, i = 1 \ldots N \} \), for e.g words embeddings. Consider similarly \( q \in \Delta_M \) representing histograms whose bins are defined on a target label space \( \Omega^T = \{ y_j \in \mathbb{R}^d, j = 1 \ldots M \} \). Consider a cost function \( c(x,y) \), (for example \( c(x,y) = \| x - y \|_2^2 \)). Let \( C \) be the matrix \( C_{ij} = c(x_i, y_j) \). \( N \) denotes a vector with all ones. Let \( \gamma \in \mathbb{R}^{N \times M} \) be a coupling matrix whose marginals are \( p \) and \( q \) such that: \( \gamma \in \Pi(p,q) = \{ \gamma \in \mathbb{R}^{N \times M}, \gamma_{1:M} = p, \gamma^T 1_N = q \} \). The optimal transport metric is defined as follows:

\[
W(p,q) = \min_{\gamma \in \Pi(p,q)} \{ (C,\gamma) = \sum_{ij} C_{ij} \gamma_{ij} \}.
\]

When \( c(x,y) = \| x - y \|_2^2 \), this distance corresponds to the so called Wasserstein–2 distance \( W_2^2 \).

Unbalanced OT. When \( p \) and \( q \) are unnormalized and have different total masses, optimal transport metric have been extended to deal with this unbalanced case. The main idea is in relaxing the \( \lambda > 0 \) for a discrete space (word embeddings for instance) \( \Omega \).

Balanced W. Barycenters: Normalized predictions. The W. barycenter (Agueh & Carlier, 2011) of normalized predictions is defined as follows: \( \bar{\mu}_w = \arg \min_{\mu \in \mathbb{R}^N} \sum_{\ell=1}^{m} \lambda_\ell W(\rho, \mu_\ell) \), for the Wasserstein distance \( W \) defined in equation (1). Hence one needs to solve the following problem, for \( m \) coupling matrices : 

\[
\min_{\rho} \gamma_{\ell} \in \Pi(\mu_{\ell},\rho), \ell = 1 \ldots m \sum_{\ell=1}^{m} \lambda_\ell \langle C_\ell, \gamma_{\ell} \rangle.
\]

Unbalanced W. Barycenters: Unnormalized predictions. Similarly the W. barycenter of unnormalized predictions is defined as follows: \( \bar{\mu}_w = \arg \min_{\mu \in \mathbb{R}^N} \sum_{\ell=1}^{m} \lambda_\ell W_{unb}(\rho, \mu_\ell) \), for the Generalized Wasserstein distance \( W_{unb} \) defined in equation (2). Hence the unbalanced W. barycenter problem (Chizat et al., 2018) amounts to solving : for \( m \) coupling matrices \( \gamma_{\ell}, \ell = 1 \ldots m \):

\[
\min_{\rho} \min_{\gamma_{\ell} \in \Pi(\mu_{\ell},\rho), \ell = 1 \ldots m} \sum_{\ell=1}^{m} \lambda_\ell \left( \langle C_\ell, \gamma_{\ell} \rangle + \lambda \overline{\text{KL}}(\gamma_{\ell 1:M}, \mu_\ell) + \lambda \overline{\text{KL}}(\gamma_{\ell}^T 1_N, \rho) \right).
\]

3.2 Balanced and Unbalanced Wasserstein in Models Ensembling

Throughout the paper we consider \( m \) discrete prediction vectors \( \mu_{\ell} \in \mathbb{R}^N, \ell = 1 \ldots m \) defined on a discrete space (word embeddings for instance) \( \Omega^S = \{ x_i \in \mathbb{R}^d, i = 1 \ldots N \} \). We refer to \( \Omega^S \) as source spaces. Our goal is to find a consensus prediction \( \bar{\mu}_w \) defined on a discrete target space \( \Omega^T = \{ y_j \in \mathbb{R}^d, j = 1 \ldots M \} \). Let \( C_{\ell} \in \mathbb{R}^{N \times M} \) be the cost matrices, \( C_{\ell,i,j} = c(x_i, y_j) \).

Balanced W. Barycenters: Normalized predictions. The W. barycenter (Agueh & Carlier, 2011) of normalized predictions is defined as follows: \( \bar{\mu}_w = \arg \min_{\mu} \sum_{\ell=1}^{m} \lambda_\ell W(\rho, \mu_\ell) \), for the Wasserstein distance \( W \) defined in equation (1). Hence one needs to solve the following problem, for \( m \) coupling matrices : 

\[
\min_{\rho} \gamma_{\ell} \in \Pi(\mu_{\ell},\rho), \ell = 1 \ldots m \sum_{\ell=1}^{m} \lambda_\ell \langle C_\ell, \gamma_{\ell} \rangle.
\]

Unbalanced W. Barycenters: Unnormalized predictions. Similarly the W. barycenter of unnormalized predictions is defined as follows: \( \bar{\mu}_w = \arg \min_{\mu} \sum_{\ell=1}^{m} \lambda_\ell W_{unb}(\rho, \mu_\ell) \), for the Generalized Wasserstein distance \( W_{unb} \) defined in equation (2). Hence the unbalanced W. barycenter problem (Chizat et al., 2018) amounts to solving : for \( m \) coupling matrices : 

\[
\min_{\rho} \min_{\gamma_{\ell} \in \Pi(\mu_{\ell},\rho), \ell = 1 \ldots m} \sum_{\ell=1}^{m} \lambda_\ell \left( \langle C_\ell, \gamma_{\ell} \rangle + \lambda \overline{\text{KL}}(\gamma_{\ell 1:M}, \mu_\ell) + \lambda \overline{\text{KL}}(\gamma_{\ell}^T 1_N, \rho) \right).
\]

3.3 Computation via Entropic Regularization and Practical Advantages

Entropic Regularized Wasserstein Barycenters Algorithms. The computation of the Wasserstein distance grows super-cubicly in the number of points. This issue was alleviated by the introduction of the entropic regularization (Cuturi, 2013) to the optimization problem making it strongly convex. Its solution can be found using scaling algorithms such as the so called Sinkhorn algorithm. For any positive matrix \( \gamma \), the entropy is defined as follows: \( H(\gamma) = -\sum_{ij} \gamma_{ij} (\log(\gamma_{ij}) - 1) \). The entropic regularized OT distances in the balanced and unbalanced case become, for a hyper-parameter \( \varepsilon > 0 \):

\[
W_\varepsilon(p,q) = \min_{\gamma \in \Pi(p,q)} \langle C,\gamma \rangle - \varepsilon H(\gamma),
\]

\[
W_{unb,\varepsilon}(p,q) = \min_{\gamma} \langle C,\gamma \rangle + \lambda \overline{\text{KL}}(\gamma_{1:M},p) + \lambda \overline{\text{KL}}(\gamma^T 1_N, q) - \varepsilon H(\gamma)
\]
for $\varepsilon \to 0$, $W_\varepsilon$ and $W_{\text{unb},\varepsilon}$ converge to the original OT distance, and for higher value of $\varepsilon$ we obtain the so called Sinkhorn divergence that allows for more diffuse transport between $p$ and $q$. Balanced and unbalanced W. barycenters can be naturally defined with the entropic regularized OT distance as follows: $\min_p \sum_{\ell=1}^M \lambda_\ell W_\varepsilon(\rho, \mu_\ell)$ and $\min_p \sum_{\ell=1}^M \lambda_\ell W_{\text{unb},\varepsilon}(\rho, \mu_\ell)$ respectively. This regularization leads to simple iterative algorithms (Benamou et al., 2015; Chizat et al., 2018) (for more details we refer the interested reader to (Chizat et al., 2018) and references therein) for computing W. barycenters that are given in Algorithms 1 and 2.

We see from this that $\lambda_\ell$ appears as matrix product multiplying individual models probability $\mu_\ell$ and the quantities $v_\ell^T$ related to Lagrange multipliers. This matrix vector product with $K_\ell$ ensures probability mass transfer between semantically related classes i.e between items that has entries with high values.

**Remark 1** (The case $K_\ell = K = I$). As the kernel $K$ in Algorithm 1 approaches I (identity) (this happens when $\varepsilon \to 0$), the alternating Bregman projection of (Benamou et al., 2015) for balanced W. barycenter converges to the geometric mean $\mu_q = \prod_{\ell=1}^M (\mu_\ell)^{\lambda_\ell}$. We prove this in Appendix D. When $K = I$ the fixed point of Algorithm 1 reduces to geometric mean, and hence diverges from the W. barycenter. Note that $K$ approaches identity as $\varepsilon \to 0$, and in this case we don’t exploit any semantics.

**Wasserstein Ensembling in Practice.** Table 1 gives a summary of machine learning tasks that can benefit from Wasserstein Ensembling, and highlights the source and target domains as well as the corresponding kernel matrix $K$. In the simplest case $\Omega_\beta^K = \Omega^T$ and $N_\ell = M$ for all $\ell$, this corresponds to the case we discussed in multi-class and multi-labels ensemble learning. W. barycenters allows to balance semantics and confidence in finding the consensus. The case where source and target spaces are different is also of interest, we give here an application example in attribute based classification: $\mu_\ell$ corresponds to prediction on a set of attributes and we wish to make predictions through the W. barycenter on a set of labels defined with those attributes. See Section 6.1. Table 12 in Appendix C gives other possible machine learning applications beyond the one explored in this paper. Appendix E discusses the computational complexity of Alg. 1 and 2.

4 **THEORETICAL ADVANTAGES OF WASSERSTEIN BARYCENTERS IN MODELS ENSEMBLING**

**Smoothness of the Wasserstein Barycenter within Semantically Coherent Clusters.** In this section we consider $\Omega_\beta^K = \Omega_\ell^K = \Omega, \text{ i.e the W. barycenters and all individual models are defined}$ on the same label space. When we are ensembling models, one desiderata is to have an accurate aggregate model. **Smoothness and Diversity** of the predictions of the ensemble is another desiderata as we often want to supply many diverse hypotheses. In the context of sequence generation in language modeling such as image captioning, machine translation or dialog, this is very important...
Table 1: Machine learning tasks where W. Ensembling can be applied. Note that W. barycenter allows ensembling different source domains to another target domain as in attributes to category.

as we use beam search on the predictions, diversity and smoothness of the predictions become key to the creativity and the composition of the sequence generator in order to go beyond "baby talk" and vanilla language based on high count words in the training set. Hence we need to increase the entropy of the prediction by finding a semantic consensus whose predictions are diverse and smooth on semantically coherent concepts without compromising accuracy. We will show in the following proposition that the W. barycenter allows such aggregation:

**Proposition 1** (Properties of Wasserstein Barycenters). Let \( \nu \) be the target distribution (an oracle) defined on a discrete space \( \Omega = \{x_1, \ldots, x_K\} \) (word embedding space) and \( \mu_\ell, \ell = 1 \ldots m \) be \( m \) estimates of \( \nu \). Assume \( W_2^2(\mu_\ell, \nu) \leq \varepsilon_\ell \). The W. barycenter \( \bar{\mu}_w \) of \( \{\mu_\ell\} \) satisfies the following:

1) **Semantic Accuracy (Distance to an oracle).** We have: \( W_2^2(\bar{\mu}_w, \nu) \leq 4 \sum_{\ell=1}^m \lambda_\ell W_2^2(\mu_\ell, \nu) \). Assume that \( W_2^2(\mu_\ell, \nu) \leq \varepsilon_\ell \), then we have: \( W_2^2(\bar{\mu}_w, \nu) \leq 4 \sum_{\ell=1}^m \lambda_\ell \varepsilon_\ell \).

2) **Diversity.** The diversity of the W. barycenter depends on the diversity of the models with respect to the Wasserstein distance (pairwise Wasserstein distance between models): \( W_2^2(\bar{\mu}_w, H_k) \leq \sum_{\ell \neq k}^m \lambda_\ell W_2^2(\mu_\ell, \mu_k) \), \( \forall k = 1, \ldots, m \).

3) **Smoothness in the embedding space.** Define the smoothness energy \( \mathcal{E}(\rho) = \sum_{ij} \|x_i - x_j\|^2 \rho_i \rho_j \). We have: \( \mathcal{E}(\bar{\mu}_w) \leq \sum_{\ell=1}^m \lambda_\ell \mathcal{E}(\mu_\ell) \). The W. barycenter is smoother in the embedding space than the individual models.

4) **Entropy.** Let \( H(\rho) = -\sum_{i=1}^K \rho_i \log(\rho_i) \), we have: \( H(\bar{\mu}_w) \geq \sum_{\ell=1}^m \lambda_\ell H(\mu_\ell) \).

**Proof.** The proof is given in Appendix F. \( \square \)

We see from Proposition 1 that the W. barycenter preserves accuracy, but has a higher entropy than the individual models. This entropy increase is due to an improved smoothness on the embedding space: words that have similar semantics will have similar probability mass assigned in the barycenter. The diversity of the barycenter depends on the Wasserstein pairwise distance between the models: the W. barycenter output will be less diverse if the models have similar semantics as measured by the Wasserstein distance. The proof of proposition 1 relies on the notion of convexity along generalized geodesics of the Wasserstein 2 distance (Agueh & Carlier, 2011). Propositions 2 and 3 in Appendix F give similar results for geometric and arithmetic mean, note that the main difference is that the guarantees are given in terms of \( K \ell \) and \( \ell_2 \) respectively, instead of \( W_2 \).

In order to illustrate the diversity and smoothness of the W. barycenter, we give here a few examples of the W. barycenter on a vocabulary of size 10000 words, where the cost matrix is constructed from word synonyms ratings, defined using Power Thesaurus or using GloVe word embeddings (Pennington et al., 2014). We compute the W. barycenter (using Algorithm 1) between softmax outputs of 4 image captioners trained with different random seeds and objective functions. Figure 4 shows the W. barycenter as well as the arithmetic and geometric mean. It can be seen that the W. barycenter has higher entropy and is smooth along semantics (synonyms or semantics in the GloVe space) and hence more diverse than individual models. Table 2 shows top 15 words of barycenter, arithmetic and geometric means, from which we see that indeed the W. barycenter outputs clusters according to semantics. In order to map back the words \( x_i \) that have high probability in the W. barycenter to an individual model \( \ell \), we can use the couplings \( \gamma^\ell \) as follows: \( \gamma^\ell_{ij} \) is the coupling between word \( j \) in the barycenter and word \( i \) in model \( \ell \). Examples are given in supplement in Table 9.
Controllable Entropy via Regularization. As the entropic regularization parameter $\varepsilon$ increases the distance of the kernel $K$ from identity $I$ increases and the entropy of the optimal couplings $\gamma_\varepsilon (H(\gamma))$ increases as well. Hence the entropy of entropic regularized $W$. Barycenter is controllable via the entropic regularization parameter $\varepsilon$. In fact since the barycenter can be written as $\mu_{\varepsilon} = \gamma_\varepsilon 1_{N,\varepsilon}$, one can show that (Lemma 2 in Appendix):

$$H(\mu_{\varepsilon}) + H(\mu_{\varepsilon}) \geq -\sum_{i,j} \gamma_{ij}^\varepsilon \log(\gamma_{ij}^\varepsilon), \forall \ell = 1 \ldots m,$$

As epsilon increases the right-hand side of the inequality increases and so does $H(\mu_{\varepsilon})$. This is illustrated in Tables 3 and 8, we see that the entropy of the (entropic regularized) $W$. barycenter increases as the distance of the kernel $K$ to identity increases ($\|K-I\|_F$ increases as $\varepsilon$ increases) and the output of the $W$. barycenter remains smooth within semantically coherent clusters.

<table>
<thead>
<tr>
<th>Rank</th>
<th>W. Barycenter</th>
<th>Arithmetic</th>
<th>Geometric</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
<th>Model 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>car 0.55</td>
<td>car 0.54</td>
<td>car 0.55</td>
<td>car 0.54</td>
<td>car 0.55</td>
<td>car 0.55</td>
<td>car 0.55</td>
<td>car 0.55</td>
</tr>
<tr>
<td>2</td>
<td>van 0.39</td>
<td>car 0.40</td>
<td>car 0.41</td>
<td>car 0.40</td>
<td>car 0.41</td>
<td>car 0.41</td>
<td>car 0.41</td>
<td>car 0.41</td>
</tr>
<tr>
<td>3</td>
<td>truck 0.39</td>
<td>truck 0.40</td>
<td>truck 0.40</td>
<td>truck 0.40</td>
<td>truck 0.40</td>
<td>truck 0.40</td>
<td>truck 0.40</td>
<td>truck 0.40</td>
</tr>
<tr>
<td>4</td>
<td>vehicle 0.46</td>
<td>vehicle 0.46</td>
<td>vehicle 0.46</td>
<td>vehicle 0.46</td>
<td>vehicle 0.46</td>
<td>vehicle 0.46</td>
<td>vehicle 0.46</td>
<td>vehicle 0.46</td>
</tr>
<tr>
<td>5</td>
<td>wagon 0.35</td>
<td>bus 0.35</td>
<td>bus 0.35</td>
<td>bus 0.35</td>
<td>bus 0.35</td>
<td>bus 0.35</td>
<td>bus 0.35</td>
<td>bus 0.35</td>
</tr>
<tr>
<td>6</td>
<td>automob 0.32</td>
<td>train 0.32</td>
<td>train 0.32</td>
<td>train 0.32</td>
<td>train 0.32</td>
<td>train 0.32</td>
<td>train 0.32</td>
<td>train 0.32</td>
</tr>
<tr>
<td>7</td>
<td>coach 0.29</td>
<td>vehicle 0.29</td>
<td>vehicle 0.29</td>
<td>vehicle 0.29</td>
<td>vehicle 0.29</td>
<td>vehicle 0.29</td>
<td>vehicle 0.29</td>
<td>vehicle 0.29</td>
</tr>
<tr>
<td>8</td>
<td>bus 0.28</td>
<td>truck 0.28</td>
<td>truck 0.28</td>
<td>truck 0.28</td>
<td>truck 0.28</td>
<td>truck 0.28</td>
<td>truck 0.28</td>
<td>truck 0.28</td>
</tr>
<tr>
<td>9</td>
<td>sedan 0.27</td>
<td>sedan 0.27</td>
<td>sedan 0.27</td>
<td>sedan 0.27</td>
<td>sedan 0.27</td>
<td>sedan 0.27</td>
<td>sedan 0.27</td>
<td>sedan 0.27</td>
</tr>
<tr>
<td>10</td>
<td>cab 0.25</td>
<td>photo 0.25</td>
<td>photo 0.25</td>
<td>photo 0.25</td>
<td>photo 0.25</td>
<td>photo 0.25</td>
<td>photo 0.25</td>
<td>photo 0.25</td>
</tr>
<tr>
<td>11</td>
<td>wheel 0.25</td>
<td>wheel 0.25</td>
<td>wheel 0.25</td>
<td>wheel 0.25</td>
<td>wheel 0.25</td>
<td>wheel 0.25</td>
<td>wheel 0.25</td>
<td>wheel 0.25</td>
</tr>
<tr>
<td>12</td>
<td>picture 0.19</td>
<td>picture 0.19</td>
<td>picture 0.19</td>
<td>picture 0.19</td>
<td>picture 0.19</td>
<td>picture 0.19</td>
<td>picture 0.19</td>
<td>picture 0.19</td>
</tr>
<tr>
<td>13</td>
<td>jeep 0.18</td>
<td>jeep 0.18</td>
<td>jeep 0.18</td>
<td>jeep 0.18</td>
<td>jeep 0.18</td>
<td>jeep 0.18</td>
<td>jeep 0.18</td>
<td>jeep 0.18</td>
</tr>
</tbody>
</table>

Table 2: Sample output (top 15 words) of $W$. barycenter (Algorithm 1), arithmetic and geometric means based on four captioner models. Each row shows a word and a corresponding probability over the synonyms and words related to the top word “car” and downweights the irrelevant objects (exploiting the side information $K$). Simple averaging techniques, which use only the confidence information, mimic the original model outputs. Figure 4 in Appendix gives a histogram view.

# RELATED WORK

Wasserstein Barycenters in Machine Learning. Optimal transport is a relatively new comor to the machine learning community. The entropic regularization introduced in (Cuturi, 2013) fostered many applications and computational developments. Learning with a Wasserstein loss in a multi-label setting was introduced in (Frogner et al., 2015), representation learning via the Wasserstein discriminant analysis followed in (Flamary et al., 2016). More recently a new angle on generative adversarial networks learning with the Wasserstein distance was introduced in (Arjovsky et al., 2017; Genevay et al., 2017; Salimans et al., 2018). Applications in NLP were pioneered by the work on Word Mover Distance (WMD) on word embeddings of (Kusner et al., 2015). Thanks to new algorithmic developments (Cuturi & Doucet, 2014; Benamou et al., 2015) $W$. barycenters have been applied to various problems: in graphics (Solomon et al., 2015), in clustering (Ye et al., 2017), in dictionary learning (Schmitz et al., 2018), in topic modeling (Xia et al., 2018), in bayesian averaging (Rios et al., 2018), and in learning word and sentences embeddings (Muzellec & Cuturi, 2018; Pal...
Singh et al., 2018) etc. Most of these applications of W. barycenter focus on learning balanced barycenters in the embedding space (like learning the means of the clusters in clustering), in our ensembling application we assume the embeddings given to us (such as GloVe word embedding) and compute the barycenter at the predictions level. Finally incorporating side information such as knowledge graphs or word embeddings in classification is not new and has been exploited in diverse ways at the level of individual model training via graph neural networks (Marino et al., 2017; Deng et al., 2014), in the framework of W. barycenter we use this side information at the ensemble level.

6 Applications

In this Section we evaluate W. barycenter ensembling in the problems of attribute-based classification, multi-label prediction and in natural language generation in image captioning.

6.1 Attribute Based Classification

As a first simple problem we study object classification based on attribute predictions. We use Animals with Attributes (Xian et al., 2017) which has 85 attributes and 50 classes. We have in our experiments 2 attributes classifiers to predict the absence/presence of each of the 85 attributes independently, based on (1) resnet18 and (2) resnet34 (He et al., 2016) input features while training only the linear output layer (following the details in Section 6.2). We split the data randomly in 30322 / 3500 / 3500 images for train / validation / test respectively. We train the attribute classifiers on the train split.

Based on those two attributes detectors we would like to predict the 50 categories using unbalanced W. barycenters using Algorithm 2. Note that in this case the source domain is the set of the 85 attributes and the target domain is the set of 50 animal categories. For Algorithm 2 we use a column-normalized version of the binary animal/attribute matrix as $K$ matrix (85 $\times$ 50), such that per animal the attribute indicators sum to 1. We selected the hyperparameters $\varepsilon = 0.3$ and $\lambda = 2$ on the validation split and report here the accuracies on the test split.

<table>
<thead>
<tr>
<th></th>
<th>resnet18 alone</th>
<th>resnet34 alone</th>
<th>Arithmetic</th>
<th>Geometric</th>
<th>W. Barycenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation</td>
<td>0.7771</td>
<td>0.8280</td>
<td>0.8129</td>
<td>0.8123</td>
<td>0.8803</td>
</tr>
<tr>
<td>Test</td>
<td>0.7714</td>
<td>0.8171</td>
<td>0.8071</td>
<td>0.8060</td>
<td>0.8680</td>
</tr>
</tbody>
</table>

Table 4: Attribute-based classification. The W. barycenter ensembling achieves better accuracy by exploiting the cross-domain similarity matrix $K$, compared to a simple linear-transform of probability mass from one domain to another as for the original models or their simple averages.

As a baseline for comparison, we use arithmetic mean ($\bar{\mu}_a$) and geometric mean ($\bar{\mu}_g$) ensembling of the two attribute classifiers resnet18 and resnet34. Then, using the same matrix $K$ as above, we define the probability of category $c$ (animal) as $p(c|\mu) = K^T \bar{\mu}$ (for $\bar{\mu} = \bar{\mu}_a$ and $\bar{\mu}_g$ resp.). We see from Table 4 that W. barycenter outperforms arithmetic and geometric mean on this task and shows its potential in attribute based classification.

6.2 Multi-label Prediction

For investigating W. barycenters on a multi-label prediction task, we use MS-COCO (Lin et al., 2014) with 80 objects categories. MS-COCO is split into training (≈82K images), test (≈35K), and validation (5K) sets, following the Karpathy splits used in the community (Karpathy & Li, 2015b).

From the training data, we build a set of 8 models using ‘resnet18’ and ‘resnet50’ architectures (He et al., 2016). To ensure some diversity, we start from pretrained models from either ImageNet (Deng et al., 2009) or Places365 (Zhou et al., 2017). Each model has its last fully-connected (‘fc’) linear layer replaced by a linear layer allowing for 80 output categories. All these pretrained models are fine-tuned with some variations. The ‘fc’ layer is trained for all models, some also fine-tune the rest of the model, while some fine-tune only the ‘layer4’ of the ResNet architecture. These variations are summarized in Table 5. Training of the ‘fc’ layer uses a $10^{-3}$ learning rate, while all fine-tunings use $10^{-6}$ learning rate. All multi-label trainings use ADAM (Kingma & Ba, 2015) with ($\beta_1 = 0.9, \beta_2 = 0.999$) for learning rate management and are stopped at 40 epochs. Only the center crop of 224$\times$224 of an input image is used once its largest dimension is resized to 256.
Table 5: Description of our 8 models built on MS-COCO

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Pretraining</th>
<th>Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>resnet18</td>
<td>ImageNet</td>
<td>fc only</td>
</tr>
<tr>
<td></td>
<td>Places365</td>
<td>fc + fine-tuning</td>
</tr>
<tr>
<td>resnet50</td>
<td>ImageNet</td>
<td>fc + layer4 fine-tuning</td>
</tr>
<tr>
<td></td>
<td>Places365</td>
<td>r50.plc.fc</td>
</tr>
</tbody>
</table>

**Table 6: Multi-label models performances compared to published results on MS-COCO test set.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP</td>
<td>59.8</td>
<td>59.5</td>
<td>60.1</td>
<td>60.1</td>
<td>61.8</td>
<td>61.4</td>
<td>61.6</td>
<td>58.3</td>
<td>62.0</td>
<td>61.6</td>
<td>58.3</td>
<td>62.0</td>
<td>58.3</td>
</tr>
</tbody>
</table>

**Evaluation Metric.** We use the mean Average Precision (mAP) which gives the area under the curve of \( P = f(R) \) for precision \( P \) and recall \( R \), averaged over each class. mAP performs a sweep of the threshold used for detecting a positive class and captures a broad view of a multi-label predictor performance. Performances for our 8 models are reported in Table 6. Precision, Recall and F1 for micro/macro are given in Table 10. Our individual models have reasonable performances overall.

Arithmetic and geometric means offer direct mAP improvements over our 8 individual models. For unbalanced W. barycenter, the transport of probability mass is completely defined by its matrix \( K = K_t \) in Algorithm 2. We investigated multiple \( K \) matrix candidates by defining \( K(i,j) \) as (i) the pairwise GloVe distance between categories, (ii) pairwise visual word2vec embeddings distance, (iii) pairwise co-occurrence counts from training data. In our experience, it is challenging to find a generic \( K \) that works well overall. Indeed, W. barycenter will move mass exactly as directed by \( K \). A generic \( K \) from prior knowledge may assign mass to a category that may not be present in some images at test time, and get harshly penalized by our metrics. A successful approach is to build a diagonal \( K \) for each test sample based on the top-N scoring categories from each model and assign the average of model posteriors scores \( K(i,i) = \frac{1}{M} \sum_m P_m(i|x) \) for image \( x \) and category \( i \). If a category is not top scoring, a low \( K(i,i) = \zeta \) value is assigned to it, diminishing its contribution. It gives W. barycenter the ability to suppress categories not deemed likely to be present, and reinforce the contributions of categories likely to be. This simple diagonal \( K \) gives our best results when using the top-2 scoring categories per model (the median number of active class in our training data is about 2) and outperforms arithmetic and geometric means as seen in Table 6. In all our experiments, W. barycenters parameters \( \{\varepsilon, \lambda\} \) in Algorithm 2 and \( \zeta \) defined above were tuned on validation set (5K). We report results on MS-COCO test set (\( \approx 35K \)). In this task of improving our 8 models, W. barycenter offers a solid alternative to commonly used arithmetic and geometric means. Appendix B.2 shows that non-uniform weighting further improves W. ensembling performance.

### 6.3 Image Captioning

In this task the objective is to find a semantic consensus by ensembling 5 image captioner models. The base model is an LSTM-based architecture augmented with the attention mechanism over the image. In this evaluation we selected captioners trained with cross entropy objective as well as GAN-trained models (Dognin et al., 2018). The training was done on COCO dataset (Lin et al., 2014) using data splits from (Karpathy & Li, 2015a): training set of 113k images with 5 captions each, 5k validation set, and 5k test set. The size of the vocabulary size is 10096 after pruning words with counts less than 5. The matrix \( K_t = K \) in Algorithm 1 was constructed using word similarities, defined based on (i) GloVe word embeddings, so that \( K = \exp(-C/\varepsilon) \), where cost matrix \( C \) is constructed based on euclidean distance between normalized embedding vectors; and (ii) synonym relationships, where we created \( K \) based on the word synonyms graph and user votes from Power Thesaurus. The model prediction \( \mu_\ell \), for \( \ell = 1, \ldots, 5 \) was selected as the softmax output of the captioner’s LSTM at the current time step, and each model’s input was weighted...
equally: $\lambda_i = 1/m$. Once the barycenter $p$ was computed, the result was fed into a beam search (beam size $B = 5$), whose output, in turn, was then given to the captioner’s LSTM and the process continued until a stop symbol (EOS) was generated. In order to exploit the controllable entropy of W. barycenter via the entropic regularization parameter $\varepsilon$, we also decode using randomized Beam search of (Shao et al., 2017), where instead of maintaining the top $k$ values, we sample $D$ candidates in each beam. The smoothness of the barycenter in semantic clusters and its controllable entropy promotes diversity in the resulting captions. We baseline the W. barycenter ensembling with arithmetic and geometric means.

![Figure 1: Comparison of the ensembling methods on COCO validation set using GloVe-based similarity matrix $K$ for 2 versions of beam search: topK (left panel) and randomized (right panel). The x-axis shows $\|K - I\|_F$, which corresponds to a different regularization parameter $\varepsilon$ (varied form 1 to 50). We can see that for topK beam search (left panel) the further $K$ is from the identity matrix, the larger the similarity neighborhood of each word, the more diverse are the generated captions (the barycenter has higher entropy), while still remaining semantically close to the ground truth. On the other hand, for randomized beam search (right panel), it is important to maintain a smaller similarity neighborhood, so that the generated sentences are not too different from the referenced ground truth.](image)

**Controllable entropy and diversity.** Figures 1 and 2 show the comparison of the ensembling methods on the validation set using topK and randomized beam search. The x-axis shows $\|K - I\|_F$, which corresponds to a different regularization $\varepsilon$ (varied form 1 to 50). We report two n-gram based metrics: CIDEr and SPICE scores, as well as the WMD (Word Mover Distance) similarity (Kusner et al., 2015), which computes the earth mover distance (the Wasserstein distance) between the generated and the ground truth captions using the GloVe word embedding vectors.

In topK beam search, as $\varepsilon$ increases, causing the entropy to go up, the exact n-grams matching metrics, i.e., CIDEr and SPICE, deteriorate while WMD remains stable. This indicates that while the barycenter-based generated sentences do not match exactly the ground truth, they still remain semantically close to it (by paraphrasing), as indicated by the stability of WMD similarity. The results of the GloVe-based barycenter on the test split of COCO dataset are shown in Table 7. In randomized beam search, the increase in entropy of the barycenter leads to a similar effect of paraphrasing but this works only up to a smaller value of $\varepsilon$, beyond which we observe a significant deterioration of the results. At that point all the words become neighbors and result in a very diffused barycenter, close to a uniform distribution. This diffusion effect is smaller for the synonyms-based $K$ since there are only a certain number of synonyms for each word, thus the maximum neighborhood is limited.

![Table 7: Performance of GloVe-based W. barycenter on COCO test split using topK beam search versus Geometric and Arithmetic ensembling. While the generated sentences based on W. barycenter do not match exactly the ground truth (lower CIDEr), they remain semantically close to it, while being more diverse (e.g., paraphrased) as indicated by the higher entropy and stable WMD.](image)
Robustness of W. Barycenter to Semantic Perturbations. Finally, the right panel of Figure 2, shows the robustness of the W. barycenter to random shuffling of the $\mu_\ell$ values, within semantically coherent clusters. Note that the size of those clusters increases as $K$ moves away from identity. The results show that barycenter is able to recover from those perturbations, employing the side information from $K$, while both the arithmetic and geometric means (devoid of such information) are confused by this shuffling, displaying a significant drop in the evaluation metrics.

![Figure 2: Left and Center Panels: Comparison of the ensembling methods on COCO validation set using synonyms-based similarity matrix with topK and randomized beam search. Right Panel: Comparison of ensembling methods when the predictions of the input models are shuffled according to the neighborhood structure defined by $K$. It can be seen that the W. Barycenter ensembling is able to recover from the word shuffling and produce better captions then the simple averaging methods, which are not able to exploit the provided side information.](image)

**Figure 2:** Left and Center Panels: Comparison of the ensembling methods on COCO validation set using synonyms-based similarity matrix with topK and randomized beam search. Right Panel: Comparison of ensembling methods when the predictions of the input models are shuffled according to the neighborhood structure defined by $K$. It can be seen that the W. Barycenter ensembling is able to recover from the word shuffling and produce better captions then the simple averaging methods, which are not able to exploit the provided side information.

Human Evaluation. We performed human evaluation on Amazon MTurk on a challenging set of images out of context of MS-COCO (Dognin et al., 2018). We compared three ensembling techniques: arithmetic, geometric and W. barycenter. For W. barycenter we used the similarity matrix $K$ defined by visual word2vec (Kottur et al., 2016). For the three models we use randomized beam search. We asked MTurkers to give a score for each caption on a scale 1-5 and choose the best captions based on correctness and detailedness. Captions examples are given in Fig. 6 (Appendix). Fig. 3 shows that W. barycenter has an advantage over the basic competing ensembling techniques.

![Figure 3: Human evaluation results. Left: Percentage of human picking best captions in terms of correctness and detail. Right: Mean Opinion Score on a scale 1 to 5.](image)

**Figure 3:** Human evaluation results. Left: Percentage of human picking best captions in terms of correctness and detail. Right: Mean Opinion Score on a scale 1 to 5.

7 Conclusion

We showed in this paper that W. barycenters are effective in model ensembling in machine learning. In the unbalanced case we showed their effectiveness in attribute based classification, as well as in improving the accuracy of multi-label classification. In the balanced case, we showed that they promote diversity and improve natural language generation by incorporating the knowledge of synonyms or word embeddings.
ACKNOWLEDGMENTS

Authors would like to thank Tommi Jaakkola, David Alvarez-Melis and Rogerio Feris for fruitful discussions. Authors thank also Satwik Kottur for sharing his visual word embeddings.

REFERENCES


Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In European Conference on Computer Vision, 2014.


A IMAGE CAPTIONING

In this Section we provide additional results for evaluating the W. barycenter on image captioning task. Figure 4 (which corresponds to Table 2 of the main paper), visualizes the word distribution of the considered ensembling methods for the input of 4 models. It is clear that with a proper choice of similarity matrix $K$, W. barycenter can create diverse, high entropy outputs.

<table>
<thead>
<tr>
<th>Rank</th>
<th>48.5 ($\epsilon = 10$)</th>
<th>39.2 ($\epsilon = 9$)</th>
<th>27.8 ($\epsilon = 7$)</th>
<th>21.7 ($\epsilon = 5$)</th>
<th>15.9 ($\epsilon = 3$)</th>
<th>4.2 ($\epsilon = 1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>car 08.09</td>
<td>car 13.63</td>
<td>car 30.39</td>
<td>car 40.43</td>
<td>car 41.87</td>
<td>car 41.94</td>
</tr>
<tr>
<td>1</td>
<td>cars 00.99</td>
<td>cars 01.36</td>
<td>cars 01.85</td>
<td>truck 02.30</td>
<td>truck 02.22</td>
<td>truck 02.23</td>
</tr>
<tr>
<td>2</td>
<td>vehicle 00.63</td>
<td>truck 00.88</td>
<td>truck 01.81</td>
<td>black 01.60</td>
<td>black 01.67</td>
<td>black 01.67</td>
</tr>
<tr>
<td>3</td>
<td>truck 00.60</td>
<td>vehicle 00.86</td>
<td>vehicle 01.30</td>
<td>cars 01.52</td>
<td>train 01.51</td>
<td>train 01.51</td>
</tr>
<tr>
<td>4</td>
<td>van 00.40</td>
<td>van 00.59</td>
<td>black 01.12</td>
<td>train 01.46</td>
<td>fashion 01.44</td>
<td>fashion 01.49</td>
</tr>
<tr>
<td>5</td>
<td>automob 00.38</td>
<td>automob 00.45</td>
<td>van 00.94</td>
<td>vehicle 01.31</td>
<td>bus 01.30</td>
<td>bus 01.30</td>
</tr>
<tr>
<td>6</td>
<td>black 00.26</td>
<td>black 00.44</td>
<td>train 00.88</td>
<td>bus 01.20</td>
<td>vehicle 01.14</td>
<td>vehicle 01.14</td>
</tr>
<tr>
<td>7</td>
<td>bus 00.26</td>
<td>bus 00.39</td>
<td>bus 00.83</td>
<td>van 01.01</td>
<td>photo 01.01</td>
<td>photo 01.01</td>
</tr>
<tr>
<td>8</td>
<td>parking 00.24</td>
<td>parking 00.37</td>
<td>parking 00.79</td>
<td>parking 00.98</td>
<td>van 01.01</td>
<td>van 01.01</td>
</tr>
<tr>
<td>9</td>
<td>vehicle 00.23</td>
<td>train 00.32</td>
<td>photo 00.65</td>
<td>photo 00.96</td>
<td>red 01.00</td>
<td>red 01.01</td>
</tr>
<tr>
<td>10</td>
<td>passeng 00.21</td>
<td>passeng 00.32</td>
<td>passeng 00.59</td>
<td>red 00.90</td>
<td>cars 00.95</td>
<td>parking 00.94</td>
</tr>
<tr>
<td>11</td>
<td>train 00.20</td>
<td>vehicle 00.27</td>
<td>red 00.53</td>
<td>fashion 00.85</td>
<td>parking 00.94</td>
<td>buildin 00.88</td>
</tr>
<tr>
<td>12</td>
<td>auto 00.19</td>
<td>photo 00.27</td>
<td>white 00.49</td>
<td>white 00.75</td>
<td>buildin 00.88</td>
<td>cars 00.81</td>
</tr>
<tr>
<td>13</td>
<td>driving 00.19</td>
<td>red 00.21</td>
<td>automob 00.44</td>
<td>buildin 00.72</td>
<td>passeng 00.71</td>
<td>passeng 00.71</td>
</tr>
<tr>
<td>14</td>
<td>photo 00.17</td>
<td>white 00.21</td>
<td>model 00.35</td>
<td>passeng 00.70</td>
<td>white 00.70</td>
<td>white 00.67</td>
</tr>
<tr>
<td>15</td>
<td>suv 00.16</td>
<td>auto 00.21</td>
<td>buildin 00.35</td>
<td>model 00.54</td>
<td>model 00.59</td>
<td>model 00.60</td>
</tr>
<tr>
<td>16</td>
<td>red 00.14</td>
<td>suv 00.21</td>
<td>silver 00.32</td>
<td>silver 00.45</td>
<td>picture 00.48</td>
<td>picture 00.49</td>
</tr>
<tr>
<td>17</td>
<td>white 00.14</td>
<td>driving 00.21</td>
<td>pickup 00.30</td>
<td>city 00.44</td>
<td>silver 00.47</td>
<td>silver 00.47</td>
</tr>
<tr>
<td>18</td>
<td>taxi 00.13</td>
<td>model 00.17</td>
<td>vehicle 00.29</td>
<td>picture 00.38</td>
<td>style 00.43</td>
<td>style 00.43</td>
</tr>
<tr>
<td>19</td>
<td>pickup 00.11</td>
<td>pickup 00.17</td>
<td>suv 00.29</td>
<td>style 00.37</td>
<td>city 00.38</td>
<td>city 00.38</td>
</tr>
</tbody>
</table>

Table 8: Sample output (top 20 words) of barycenter for different similarity matrices $K$ based on GloVe (columns titles denote the distance of $K$ from identity $\| K - I \|_F$ and corresponding $\epsilon$). Each column shows a word and its corresponding probability over the vocabulary. Note that the last column coincides with the output from geometric mean.

Table 8 shows the effect of entropic regularization $\epsilon$ on the resulting distribution of the words of W. barycenter using GloVe embedding matrix. As $K$ moves closer to the identity matrix, the entropy of barycenter decreases, leading to outputs that are close/identical to the geometric mean. On the other hand, with a large entropic regularization, matrix $K$ moves away from identity, becoming an uninformative matrix of all 1’s. This eventually leads to a uniform distribution which spreads the probability mass equally across all the words. This can be also visualized with a histogram in Figure 5, where the histograms on the bottom represent distributions that are close to uniform, which can be considered as failure cases of W. barycenter, since the image captioner in this case can only generate meaningless, gibberish captions.

In Table 9 we show a mapping from a few top words in the barycenter output (for similarity matrix $K$ based on synonyms) to the input models. In other words, each column defines the words in the input models which have the greatest influence on each of top 3 words in the barycenter output.

In Figure 6 we present a few captioning examples showing qualitative difference between the considered ensembling techniques.
Figure 4: Visualization of the word distributions of W. barycenter, arithmetic and geometric means based on four captioning models, whose input image is shown on top (one of the ground-truth human-annotated captions for this image reads: *A police car next to a pickup truck at an intersection*). The captioner generates a sentence as a sequence of words, where at each step the output is a distribution over the whole vocabulary. The top four histograms show a distribution over the vocabulary from each of the model at time $t = 3$ during the sentence generation process. The bottom three histograms show the resulting distribution over the vocabulary for the ensembles based on W. Barycenter, arithmetic and geometric means. It can be seen that the W. Barycenter produces high entropy distribution, spreading the probability mass over the synonyms of the word "car" (which is the top word in all the four models), based on the synonyms similarity matrix $K$. 
Figure 5: Visualization of the word distributions of W. barycenter for different similarity matrices $K$ based on GloVe (rows denote the distance of $K$ from identity $\|K - I\|_F$ and corresponding $\epsilon$). Large entropic regularization generates $K$ close to uninformative matrices of all 1’s. This eventually leads to a barycenter which is close to a uniform distribution spreading the probability mass almost equally across all the words.

Table 9: Mapping from a few top words in the barycenter output (for similarity matrix $K$ based on synonyms) to the input models. For each word in the left columns, the remaining columns show the contributing words and the percent of contribution.
We evaluate our models using micro and macro versions of precision, recall, and F1-measure as covered in multi-label prediction metrics study from (Wu & Zhou, 2016). For these measures, a threshold of 0.5 is commonly used to predict a label as positive in the community’s published results. Macro precision is an average of per-class precisions while micro precision is computed by computing the ratio of all true positives across all image samples over the number of all positive classes in a dataset. Therefore a macro (or per-class) precision ‘P-C’ is defined as $\frac{1}{C} \sum_i P_i$ while a micro (or overall) precision ‘P-O’ is defined as $\frac{\sum_i TP_i}{\sum_i TP_i + FP_i}$ where $TP_i$ and $FP_i$ are true and false positives respectively. Per-class and overall versions for R and F1 are defined similarly. We also employ mean Average Precision (mAP) which gives the area under the curve of $P = f(R)$ averaged over each class. Unlike P,R and F1, mAP inherently performs a sweep of the threshold used for detecting a positive class and captures a broader view of a multi-label predictor’s performance. Performances for our 8 models and previously published results are reported in Table 10 and in Table 6 in the paper. Our models have reasonable performances overall.

Figure 6: Examples of captions for several images. BA: Wasserstein Barycenter, AM: Arithmetic mean, GM: Geometric mean, GT: Ground truth.

### B Multi-label Prediction

#### B.1 Metrics used and Single Models Predictions

We evaluate our models using micro and macro versions of precision, recall, and F1-measure as covered in multi-label prediction metrics study from (Wu & Zhou, 2016). For these measures, a threshold of 0.5 is commonly used to predict a label as positive in the community’s published results. Macro precision is an average of per-class precisions while micro precision is computed by computing the ratio of all true positives across all image samples over the number of all positive classes in a dataset. Therefore a macro (or per-class) precision ‘P-C’ is defined as $\frac{1}{C} \sum_i P_i$ while a micro (or overall) precision ‘P-O’ is defined as $\frac{\sum_i TP_i}{\sum_i TP_i + FP_i}$ where $TP_i$ and $FP_i$ are true and false positives respectively. Per-class and overall versions for R and F1 are defined similarly. We also employ mean Average Precision (mAP) which gives the area under the curve of $P = f(R)$ averaged over each class. Unlike P,R and F1, mAP inherently performs a sweep of the threshold used for detecting a positive class and captures a broader view of a multi-label predictor’s performance. Performances for our 8 models and previously published results are reported in Table 10 and in Table 6 in the paper. Our models have reasonable performances overall.
Table 10: Our multi-label models performances compared to published results on MS-COCO test set. Arithmetic, geometric means and W. barycenter performances are reported as well. [1] (He et al., 2015) [2] (Zhu et al., 2017)

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP</th>
<th>F1-C</th>
<th>P-C</th>
<th>R-C</th>
<th>F1-O</th>
<th>P-O</th>
<th>R-O</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-101</td>
<td>59.8</td>
<td>55.7</td>
<td>65.8</td>
<td>51.9</td>
<td>72.5</td>
<td>75.9</td>
<td>69.5</td>
</tr>
<tr>
<td>ResNet-107</td>
<td>59.5</td>
<td>55.6</td>
<td>65.4</td>
<td>52.2</td>
<td>72.6</td>
<td>75.5</td>
<td>70.0</td>
</tr>
<tr>
<td>ResNet-101-sem</td>
<td>60.1</td>
<td>54.9</td>
<td>69.3</td>
<td>48.6</td>
<td>72.6</td>
<td>76.9</td>
<td>68.8</td>
</tr>
<tr>
<td>ResNet-SRN-att</td>
<td>61.8</td>
<td>56.9</td>
<td>67.5</td>
<td>52.5</td>
<td>73.2</td>
<td>76.5</td>
<td>70.1</td>
</tr>
<tr>
<td>ResNet-SRN</td>
<td>62.0</td>
<td>58.5</td>
<td>65.2</td>
<td>55.8</td>
<td>73.4</td>
<td>75.5</td>
<td>71.5</td>
</tr>
<tr>
<td>r50.plc.fc+ft</td>
<td>61.4</td>
<td>54.6</td>
<td>74.9</td>
<td>43.0</td>
<td>63.4</td>
<td>81.3</td>
<td>51.9</td>
</tr>
<tr>
<td>r50.plc.fc+ft</td>
<td>61.6</td>
<td>55.5</td>
<td>74.2</td>
<td>44.4</td>
<td>64.1</td>
<td>80.7</td>
<td>53.2</td>
</tr>
<tr>
<td>r18.plc.fc+ft</td>
<td>58.3</td>
<td>51.0</td>
<td>71.9</td>
<td>39.5</td>
<td>61.2</td>
<td>80.0</td>
<td>49.6</td>
</tr>
<tr>
<td>r18.plc.fc</td>
<td>52.3</td>
<td>43.8</td>
<td>71.3</td>
<td>31.6</td>
<td>55.6</td>
<td>79.8</td>
<td>42.7</td>
</tr>
<tr>
<td>r18.img.fc+ft</td>
<td>49.6</td>
<td>40.7</td>
<td>68.9</td>
<td>28.9</td>
<td>54.1</td>
<td>78.8</td>
<td>41.2</td>
</tr>
<tr>
<td>r18.img.fc</td>
<td>64.1</td>
<td>57.9</td>
<td>76.7</td>
<td>46.4</td>
<td>64.3</td>
<td>80.9</td>
<td>53.4</td>
</tr>
<tr>
<td>r50.img.fc</td>
<td>63.3</td>
<td>55.9</td>
<td>78.7</td>
<td>43.4</td>
<td>62.2</td>
<td>83.1</td>
<td>49.7</td>
</tr>
<tr>
<td>r18.img.fc</td>
<td>58.1</td>
<td>50.7</td>
<td>75.7</td>
<td>38.1</td>
<td>58.4</td>
<td>80.5</td>
<td>45.8</td>
</tr>
</tbody>
</table>

Table 11: multi-label models ensembling mAP on MS-COCO test set (35150 images). Performance-based weighting helps both arithmetic and W.Barycenter ensembling, the latter retaining its performance vantage.

B.2 Weighted Multi-label Prediction Ensembling

Ensembling results given in Tab. 6 are using uniformly weighted models, i.e. $\lambda_\ell = \frac{1}{m}$ where $m$ is the number of models. However, in practice, arithmetic and geometric mean ensembling usually use weighted ensembles of models The weights are then optimized and established on a small validation set before being used for ensembling on a test set. A well-known embodiment of this type of approach is Adaboost (Freund & Schapire, 1999) where weights are dynamically defined at each pass of training wrt to the accuracy of base models.

Here, we follow a much simpler but similar approach by defining the performance of each model $\ell$ as the mean average precision (mAP$_\ell$) on the validation set. mAP$_\ell$ is used to define $\lambda_\ell$ such that $\lambda_\ell = \frac{\text{mAP}_\ell}{\sum_{\ell} \text{mAP}_\ell}$. $\lambda_\ell$ are then applied to the models’ scores for arithmetic, geometric mean and W.Barycenter ensembles. Tab. 11 reports mAP for each ensembling technique over the MS-COCO test set (35150 images). Note that the $\lambda_\ell$ weights definition is based on the final metric evaluation, mAP in this case. For other tasks such as classification, accuracy or any other valid metric can be employed to compute the $\lambda_\ell$ weights. It must be noted that the weights are computed with respect to the ultimate performance metric at hand. Tab. 11 reveals clearly that such approach of weighting models by their performance benefits arithmetic and W.Barycenter ensembling for this task. Both methods leverage the confidence of the underlying models and the mAP weighting of models will reinforce the contributions of better performing models. Geometric means ensembling is not significantly impacted by non-uniform $\lambda_\ell$ since it is mostly relying on consensus of the models, not their confidence. We conclude that weighting indeed helps performance and keeps a performance advantage for W. Barycenter over the alternatives arithmetic and geometric means.

<table>
<thead>
<tr>
<th>Ensembling</th>
<th>$\lambda_\ell = \frac{1}{m}$</th>
<th>$\lambda_\ell = \frac{\text{mAP}<em>\ell}{\sum</em>{\ell} \text{mAP}_\ell}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic mean</td>
<td>64.5</td>
<td>64.8</td>
</tr>
<tr>
<td>Geometric mean</td>
<td>63.9</td>
<td>63.9</td>
</tr>
<tr>
<td>W.Barycenter</td>
<td>65.1</td>
<td>65.2</td>
</tr>
</tbody>
</table>

Table 11: multi-label models ensembling mAP on MS-COCO test set (35150 images). Performance-based weighting helps both arithmetic and W.Barycenter ensembling, the latter retaining its performance vantage.
C Other Machine Learning Tasks that can be adressed in the Wasserstein Barycenter Ensembling Framework

Table 12 summarizes Machine learning tasks that could benefit from W. Ensembling, especially in the case when source and target domains are distinct.

<table>
<thead>
<tr>
<th>Source Domains (models)</th>
<th>Target Domain (barycenter)</th>
<th>Related K (OT cost matrix)</th>
<th>Attribute Geometric apply</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-class Learning (Balanced OT)</td>
<td>$\mu_i \in \Delta_k, \ell = 1 \ldots m$ Histograms of size $N$</td>
<td>$p \in \Delta_k$ Histogram of size $N$</td>
<td>$K_{ij} = \varepsilon \frac{1 - \ell_{ij}}{\ell_{ij}}$ word embedding of category $i$ (See Section 6.1 for other choices)</td>
</tr>
<tr>
<td>Multi-label Learning (Unbalanced OT)</td>
<td>$\mu_i \in [0,1]^N, \ell = 1 \ldots m$ Soft multi-labels</td>
<td>$p \in [0,1]^N$ Soft multi-label</td>
<td>$K_{ij}$ adjacency weight in a knowledge graph (See Section 6.2 for other choices)</td>
</tr>
<tr>
<td>Cost-sensitive Classification (Balanced OT)</td>
<td>$\mu_i \in \Delta_k, \ell = 1 \ldots m$ Histograms of size $N$</td>
<td>$p \in \Delta_k$ Histogram of size $N$</td>
<td>$K_{ij}$ User ratings of similarities user-defined costs for confusion of $i$ and $j$ e.g. binary matrix for synonyms (Section 6.3)</td>
</tr>
<tr>
<td>Attributes to Categories Zero (Few) shot Learning (Unbalanced OT)</td>
<td>$\mu_i \in [0,1]^N, \ell = 1 \ldots m$ Soft multi-labels: $N$ attributes</td>
<td>$p \in \Delta_k$ Histogram of size $M$ $M$ categories</td>
<td>$K_{ij}$ Presence or absence of attribute $i$ in class $j$ (See Section 6.1)</td>
</tr>
<tr>
<td>Vocabulary Expansion (Balanced OT)</td>
<td>$\mu_i \in \Delta_k, \ell = 1 \ldots m$ Histograms on a vocabulary of size $N$</td>
<td>$p \in \Delta_k$, $M &gt; N$ Histogram on a larger vocabulary</td>
<td>$K_{ij} = \varepsilon \frac{1 - \ell_{ij}}{\ell_{ij}}$, $e_i$ word-embeddings</td>
</tr>
<tr>
<td>Multi-lingual fusion and Translation (Balanced OT)</td>
<td>$\mu_i \in \Delta_k, \ell = 1 \ldots m$ Histograms on $m$ source languages of vocabulary size $N_i$ each</td>
<td>$p \in \Delta_k$ Histogram on a target language with vocabulary size $M$</td>
<td>$K_{ij} = \varepsilon \frac{1 - \ell_{ij}}{\ell_{ij}}$, $e_i^\ell$, $e_j^\ell$ multi-lingual word embeddings, $e_i^\ell$ word embeddings of source language $i$, $e_j^\ell$ word embeddings of target language</td>
</tr>
</tbody>
</table>

Table 12: Machine learning tasks where W. Barycenter ensembling can be applied: We emphasize that W. Barycenter has the advantage over alternatives such as arithmetic or geometric means in that it is able to ensemble models whose histograms are defined on different domains. Moreover, the target domain, where the barycenter is defined, can be different from the source domains. This is encountered, for instance, in the attribute-based classification, where models are defined on attributes (multi-labels) and the ensemble is defined on the set of categories. We give here additional tasks that can benefit from this flexibility of W. Barycenters.

D Convergence of Scaling algorithms for entropic reg. W. Barycenters

For W. Barycenters (Chizat et al., 2018) proves (in Theorem 4.1 of this paper) that the fixed point algorithm given in Algorithms 1 and 2 converges under mild conditions on the matrix $K$: ($K$ takes positive values), nevertheless this does not characterize the convergence as $\varepsilon$ goes to zero.

D.1 Balanced W. Barycenters

$\Gamma$ convergence of the regularized OT problem to the original OT problem when the regularization parameter vanishes was studied in (Carlier et al., 2017), but this work does provide an algorithm that guarantees such a convergence.

For balanced W. barycenters we show in the following Lemma that the fixed point of algorithm 1 converges to the geometric mean when $K_{ij} = I$ for all $\ell = 1 \ldots m$. Assuming $C_{ij}$ is bounded, it is easy to see that $K$ converges to identity as $\varepsilon$ approaches 0. Hence the fixed point of Algorithm 1 does not recover the original unregularized W. Barycenter as $\varepsilon \rightarrow 0$.

Lemma 1. For $K_{ij} = I$, $\ell = 1 \ldots m$, the fixed point of Algorithm 1 is the geometric mean $\bar{\mu}_g = \Pi_{\ell=1}^m (\mu_\ell)^{\lambda_\ell}$.

Proof. Let us show that at any iteration we have the following recurrence:

$$u_i^t = \frac{\mu_i^{t+1}}{(\bar{\mu}_g)^t}, p^t = \bar{\mu}_g, v_i^t = \left(\frac{\mu_i}{\bar{\mu}_g}\right)^{t+1},$$

all operations are element-wise.
At iteration 0, the result holds since we have:

$$u^0_\ell = \mu_\ell, p^0 = \prod_{\ell=1}^m (u^0_\ell)_{\lambda_\ell}, v^0_\ell = \frac{\prod_{\ell=1}^m (u_\ell)_{\lambda_\ell}}{\mu_\ell}$$

Assume the result holds at time $t$. Let us prove it for $t+1$, following the updates of Algorithm 1:

$$u^{t+1}_\ell = \frac{\mu_\ell}{v^{t+1}_\ell} = \left(\frac{\mu_\ell}{\bar{\mu}_\ell}\right)^{t+1}$$

$$p^{t+1} = \prod_{\ell=1}^m \left(\frac{u^{t+1}_\ell}{\mu_\ell}\right)^{\lambda_\ell}$$

$$= \prod_{\ell=1}^m \left(\frac{\mu_\ell}{\bar{\mu}_\ell}\right)^{\lambda_\ell}$$

$$= \frac{\prod_{\ell=1}^m \mu_\ell^{\lambda_\ell}}{(\bar{\mu}_\ell)^{\lambda_\ell}}$$

$$= \frac{\bar{\mu}_\ell^{t+2}}{\bar{\mu}_\ell^{\lambda_\ell}}$$ since $\sum_{\ell=1}^m \lambda_\ell = 1$

$$= \bar{\mu}_\ell.$$

$$v^{t+1}_\ell = \frac{p_{\ell+1}}{v^{t+1}_\ell} = \left(\frac{\mu_\ell}{\bar{\mu}_\ell}\right)^{\lambda_\ell} = \left(\frac{\mu_\ell}{\bar{\mu}_\ell}\right)^{t+2}.$$

QED.

For $\varepsilon > 0$, Feydy et al (Feydy et al., 2018) showed recently that the Sinkhorn divergence defines an interpolation between the MMD distance (Maximum mean discrepancy (Gretton et al., 2012)) and the Wasserstein Distance.

Hence for $\varepsilon > 0$ Algorithm 1 provides still an interesting solution that can be seen as an interpolation between the original (unregularized) Wasserstein Barycenter and the MMD Barycenter (Frechet barycenter for $d = \text{MMD}$).

D.2 UNBALANCED W. BARYCENTER

Note that:

$$\min_{\gamma} \frac{1}{\lambda} \langle C, \gamma \rangle + \tilde{\text{KL}}(\gamma_{1:M}, p) + \tilde{\text{KL}}(\gamma_{\top 1:N}, q) - \frac{\varepsilon}{\lambda} H(\gamma)$$

$$= \min_{\gamma} \tilde{\text{KL}}(\gamma_{1:M}, p) + \tilde{\text{KL}}(\gamma_{\top 1:N}, q) + \frac{1}{\lambda} (\tilde{\text{KL}}(\gamma, e^{-C/\varepsilon}) - \varepsilon \sum_{i,j} K_{ij})$$

As $\lambda$ goes to infinity this unbalanced cost converges to the Hellinger distance:

$$\min_{\gamma} \tilde{\text{KL}}(\gamma_{1:M}, p) + \tilde{\text{KL}}(\gamma_{\top 1:N}, q) = \|\sqrt{p} - \sqrt{q}\|^2,$$

Hence using $d = \frac{1}{\lambda} W_{\text{unb},\varepsilon}$ in Unbalanced Barycenters, we approach the Hellinger Frechet mean, as $\lambda \to \infty$.

E TIME COMPLEXITY OF SCALING ALGORITHMS FOR W. BARYCENTER

E.1 COMPUTATIONAL COMPLEXITY AND IMPROVEMENTS

The total time complexity of a straight-forward implementation of Algorithms 1 and 2 is $O(mN^2 \text{Maxiter})$ where Maxiter is the number of iterations, $m$ the number of models and $N$ the number of bins.
We make the following theoretical and practical remarks on how to improve this computational complexity to reach an almost linear dependency on $N$ using low rank approximation of the kernel matrix $K$, and parallelization on $m$ machines:

1. **Dependency on Maxiter**: For the number of iterations we found that Maxiter = 5 is enough for convergence, which makes most of the computational complexity dependent on $m$ and $N$.

2. **Dependency on $N$ and low rank approximation**: The main computational complexity comes from the matrix vector multiply $K \ell u$ that is of $O(N^2)$. Note that this complexity can be further reduced since the kernel matrix $K$ is often low rank. Therefore we can be written $K = \Phi \Phi^T$ where $\Phi \in \mathbb{R}^{N \times k}$, where $k \ll N$, which allows to compute this product as follows $\Phi \Phi^T u_\ell$ that has a lower complexity $O(Nk)$. $\Phi$ can be computed using Nystrom approximation or random Fourier features. Hence potentially on can get an algorithm with complexity $O(mNk)$, where $k$ has a logarithmic dependency on $N$. This was studied recently in (Altschuler et al., 2018).

3. **Dependency on $m$ and parallelization**: Regarding the dependency on $m$, as noted in (Chizat et al., 2018; Benamou et al., 2015) the algorithm is fully parallelizable which would lead to a computational complexity of $O(Nk)$ by using simply $m$ machines.

4. **GPU and Batch version**: Practically, the algorithm implemented takes advantage of matrix vector products’ speed on GPU. The algorithm can be further accelerated by computing Sinkhorn divergences in batches as pointed in (Feydy et al., 2018).

### E.2 Time Complexity in the Multi-label Wasserstein Ensembling

We evaluated the time complexity of the GPU implementation of Wasserstein Barycenters in *pytorch* on our multi-label prediction experiments using MS-COCO test set (35150 samples). Note that we used a vanilla implementation of Algorithm 2, i.e. without parallelization, batching, or low rank approximation. Results and comments for these wall clock timings can be found in Tab. 13. As it can be observed, we need to use Maxiter = 5 on a GPU-V100 to reach below 4ms/image for Wasserstein ensembling. This is not a major overhead and can be further improved as discussed previously by using parallelization, batching and low rank approximation. In Table 13, each timing was done over the whole test set; each timing repeated 5 times. We report means and standard deviations of total wall clock times for ensembling 8 models. Last column on the right is the average timing per image (in ms) for W.Barycenter. The number of W.Barycenter iterations (Maxiter) was varied from 1, 5 and 10 to show its impact. We report timing numbers over two GPU architectures, NVIDIA Tesla K80 and V100. W.Barycenters leverage the GPU while Arithmetic and Geometric means do not. Timings are of the computations of the means themselves, no data fetching or data preparation is included in these timings. As expected, the wall clock time cost for W.Barycenters is several order of magnitude higher than for Arithmetic and Geometric means. The difference of GPU does not impact the Arithmetic and Geometric means as they do not use it in our implementation. The Barycenter computation see a speed up from K80 to V100 as V100 is much better at reducing wall time for longer number of iterations.

<table>
<thead>
<tr>
<th>GPU Model</th>
<th>Maxiter</th>
<th>Arithmetic Mean test set (s)</th>
<th>Geometric Mean test set (s)</th>
<th>W.Barycenter test set (s)</th>
<th>W.Barycenter per image (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tesla-K80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.015 ± 0.001</td>
<td>0.298 ± 0.002</td>
<td>38.954 ± 0.487</td>
<td>1.108</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.015 ± 0.000</td>
<td>0.297 ± 0.001</td>
<td>152.987 ± 1.725</td>
<td>4.352</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.015 ± 0.000</td>
<td>0.296 ± 0.002</td>
<td>302.254 ± 2.470</td>
<td>8.599</td>
</tr>
<tr>
<td>Tesla-V100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.018 ± 0.002</td>
<td>0.297 ± 0.003</td>
<td>36.742 ± 0.843</td>
<td>1.045</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.016 ± 0.002</td>
<td>0.289 ± 0.009</td>
<td>135.950 ± 5.897</td>
<td>3.868</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.014 ± 0.000</td>
<td>0.278 ± 0.001</td>
<td>249.775 ± 3.119</td>
<td>7.106</td>
</tr>
</tbody>
</table>

Table 13: Timings (in s) of Wasserstein Barycenter computation compared to Arithmetic and Geometric mean computations for the MS-COCO test set (35150 samples).
F Additional Theoretical Results and Proofs

Proposition 2 (properties of Geometric Mean). The following properties hold for geometric mean:

1. **Geometric mean is the Frechet mean of $\widetilde{KL}$**. The geometric mean is the Frechet mean with respect to the extended $\widetilde{KL}$ divergence:

   $\bar{\mu}_g = \prod_{\ell=1}^m (\mu_\ell)^{\lambda_\ell} = \arg\min_{\rho} L(\rho) := \sum_{\ell=1}^m \lambda_\ell \widetilde{KL}(\rho, \mu_\ell)$

2. **Correctness guarantee of Geometric mean in $\widetilde{KL}$**. Let $\nu$ be an oracle the geometric mean satisfies:

   $\widetilde{KL}(\nu, \bar{\mu}_g) \leq \sum_{\ell=1}^m \lambda_\ell \widetilde{KL}(\nu, \mu_\ell)$

Proposition 3 (properties of Arithmetic Mean). The following properties hold for geometric mean:

1. **Arithmetic mean is the Frechet mean of $\ell_2$**.

   $\bar{\mu}_a = \sum_{\ell=1}^m \lambda_\ell \mu_\ell = \arg\min_{\rho} L(\rho) := \sum_{\ell=1}^m \lambda_\ell \|\rho - \mu_\ell\|^2$

2. **Correctness guarantee of Arithmetic mean in $\ell_2$**. Let $\nu$ be an oracle the arithmetic mean satisfies:

   $\|\nu - \bar{\mu}_g\|_2^2 \leq 2 \sum_{\ell=1}^m \lambda_\ell \|\nu - \mu_\ell\|_2^2$.

3. **Entropy** : Strong convexity of negative entropy with respect to $\ell_1$ of arithmetic mean:

   $H(\sum_{\ell} \lambda_\ell \mu_\ell) \geq \sum_{\ell=1}^m \lambda_\ell H(\mu_\ell) + \frac{1}{2} \sum_{\ell \neq k} \lambda_\ell \lambda_k \|\mu_\ell - \mu_k\|_1^2$.

Proof of Proposition 2. 1)

\[
\min_{\rho} L(\rho) := \sum_{\ell=1}^m \lambda_\ell \widetilde{KL}(\rho, \mu_\ell) = \sum_{i=1}^N \sum_{\ell=1}^m \lambda_\ell \left( \rho_i \log \left( \frac{\rho_i}{\mu_{\ell,i}} \right) - \rho_i + \mu_{\ell,i} \right)
\]

First order optimality condition:

\[
\frac{\partial L}{\partial \rho_i} = \sum_{\ell=1}^m \lambda_\ell \log \left( \frac{\rho_i}{\mu_{\ell,i}} \right) = 0,
\]

This gives us the result:

$\rho_i = \prod_{\ell=1}^m (\mu_{\ell,i})^{\lambda_\ell}$. 

22
Raising to the power 2 we have:

\[ \mathcal{KL}(\nu, \bar{\mu}_g) = \sum_i (\nu_i \log \left( \frac{\nu_i}{\Pi_{\ell=1}^m (\mu_{\ell,i})^{\lambda_{\ell}}} \right) - \nu_i + \Pi_{\ell=1}^m (\mu_{\ell,i})^{\lambda_{\ell}}) \]

\[ = \sum_i (\nu_i \log \left( \frac{\nu_i}{\Pi_{\ell=1}^m (\mu_{\ell,i})^{\lambda_{\ell}}} \right) - \nu_i + \Pi_{\ell=1}^m (\mu_{\ell,i})^{\lambda_{\ell}}) \text{ using } \sum_{\ell=1}^m \lambda_{\ell} = 1 \]

\[ = \sum_{\ell=1}^m \lambda_{\ell} \left( \sum_i (\nu_i \log \left( \frac{\nu_i}{\mu_{\ell,i}} \right) - \nu_i) + \Pi_{\ell=1}^m (\mu_{\ell,i})^{\lambda_{\ell}} \right) \]

\[ = \sum_{\ell=1}^m \lambda_{\ell} \left( \sum_i (\nu_i \log \left( \frac{\nu_i}{\mu_{\ell,i}} \right) - \nu_i + \mu_{\ell,i}) \right) + \sum_{i} \left( \Pi_{\ell=1}^m (\mu_{\ell,i})^{\lambda_{\ell}} - \sum_{\ell=1}^m \lambda_{\ell} \mu_{\ell,i} \right) \]

\[ \leq \sum_{\ell=1}^m \lambda_{\ell} \mathcal{KL}(\nu, \mu_{\ell}), \]

The last inequality follows from the arithmetic-geometric mean inequality (Jensen inequality):

\[ \Pi_{\ell=1}^m (\mu_{\ell,i})^{\lambda_{\ell}} - \sum_{\ell=1}^m \lambda_{\ell} \mu_{\ell,i} \leq 0. \]

3)

\[ \left| \mathcal{KL}(\nu, \bar{\mu}_g) - \sum_{\ell=1}^m \lambda_{\ell} \mathcal{KL}(\nu, \mu_{\ell}) \right| \leq \| \bar{\mu}_g - \bar{\mu}_a \|_1 \]

\[ \varepsilon_{\ell} - \| \bar{\mu}_g - \bar{\mu}_a \|_1 \leq \mathcal{KL}(\nu, \bar{\mu}_g) \leq \varepsilon_{u} + \| \bar{\mu}_g - \bar{\mu}_a \|_1 \]

\[ \square \]

**Proof of Proposition 1.** 1) By the triangle inequality we have for all \( \ell \):

\[ W_2(\bar{\mu}_w, \nu) \leq W_2(\bar{\mu}_w, \mu_{\ell}) + W_2(\mu_{\ell}, \nu) \]

Raising to the power 2 we have:

\[ W_2^2(\bar{\mu}_w, \nu) \leq (W_2(\bar{\mu}_w, \mu_{\ell}) + W_2(\mu_{\ell}, \nu))^2 \leq 2(W_2^2(\bar{\mu}_w, \mu_{\ell}) + W_2^2(\mu_{\ell}, \nu)), \]

where we used \((a + b)^2 \leq 2(a^2 + b^2)\). Summing over all \( \ell \), and using \( \sum_{\ell=1}^m \lambda_{\ell} = 1 \) we have:

\[ W_2^2(\bar{\mu}_w, \nu) \leq 2 \left( \sum_{\ell=1}^m \lambda_{\ell} W_2^2(\bar{\mu}_w, \mu_{\ell}) + \sum_{\ell=1}^m \lambda_{\ell} W_2^2(\mu_{\ell}, \nu) \right) \]

\[ \leq 2 \left( \sum_{\ell=1}^m \lambda_{\ell} W_2^2(\nu, \mu_{\ell}) + \sum_{\ell=1}^m \lambda_{\ell} W_2^2(\mu_{\ell}, \nu) \right) \]

( By Barycenter definition \( \sum_{\ell=1}^m \lambda_{\ell} W_2^2(\bar{\mu}_w, \mu_{\ell}) \leq \sum_{\ell=1}^m \lambda_{\ell} W_2^2(\nu, \mu_{\ell}) \).

\[ = 4 \sum_{\ell=1}^m \lambda_{\ell} W_2^2(\nu, \mu_{\ell}) (\text{Using the symmetry of } W_2). \]

2) For a fixed base distribution \( \mu \) the functional \( W_2^2(\cdot, \mu) \) is convex along generalized geodesics (Santambrogio, May 2015). Given two distributions \( \nu_1, \nu_2 \), and \( T_1 \) and \( T_2 \), such that \( T_1, # \mu = \nu_1 \) and \( T_2, # \mu = \nu_2 \), we have:

\[ W_2^2((tT_1 + (1-t)T_2)\mu, \mu) \leq tW_2^2(\nu_1, \mu) + (1-t)W_2^2(\nu_2, \mu). \]
Fix $k \in [1, m]$. Note that there exists $T^k_\ell$, $\ell = 1 \ldots m$, such that $\bar{\mu}_w = (\sum_\ell \lambda_\ell T^k_\ell) \# \mu_k$, where $T^k_\ell \# \mu_k = \mu_\ell$. Hence we have:

$$W^2_2(\bar{\mu}_w, \mu_k) = W^2_2\left(\left(\sum_\ell \lambda_\ell T^k_\ell \right) \# \mu_k, \mu_k\right) \leq \sum_{\ell=1}^m \lambda_\ell W^2_2(T^k_\ell \# \mu_k, \mu_k) = \sum_{\ell \neq k} \lambda_\ell W^2_2(\mu_\ell, \mu_k).$$

3) The smoothness energy is a valid interaction energy, by Proposition 7.7 Point 3 (for $W = \|\cdot\|^2$) (Agueh & Carlier, 2011), it is convex along generalized geodesics and hence by Proposition 7.6 it is barycenter convex:

$$E(\bar{\mu}_w) \leq \sum_{\ell=1}^m \lambda_\ell E(\mu_\ell).$$

4) The negative entropy $(-H(\rho))$ is a valid internal energy, by Proposition 7.7 Point 1 ($F(t) = -t \log(t)$) (Agueh & Carlier, 2011), it is convex along generalized geodesics and hence by Proposition 7.6 it is barycenter convex:

$$-H(\bar{\mu}_w) \leq -\sum_{\ell=1}^m \lambda_\ell H(\mu_\ell).$$

**Remark 2** (Arithmetic mean and negative entropy). The negative entropy on discrete space is convex and moreover it is strongly convex with respect to the $\ell^1$ norm hence we have:

$$-H\left(\sum \lambda_\ell \mu_\ell\right) \leq -\sum_{\ell=1}^m \lambda_\ell H(\mu_\ell) - \frac{1}{2} \sum_{\ell \neq k} \lambda_\ell \lambda_k \|\mu_\ell - \mu_k\|_1^2,$$

hence the entropy of the arithmetic mean satisfies:

$$H\left(\sum \lambda_\ell \mu_\ell\right) \geq \sum_{\ell=1}^m \lambda_\ell H(\mu_\ell) + \frac{1}{2} \sum_{\ell \neq k} \lambda_\ell \lambda_k \|\mu_\ell - \mu_k\|_1^2.$$

hence the diversity of the arithmetic mean depends on the $\ell_1$ distance between the individual models, that does not take in count any semantics.

**Remark 3.** Recall strong convexity of $f$ in $\mathbb{R}^d$, there exists $\kappa > 0$:

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) - (t)(t-1)\kappa \|x - y\|_2^2$$

To show something about how the entropy of barycenter depends on pairwise distances we need something on “strong geodesic convexity” or “strong barycenter convexity” of negative entropy (Conjectured here):

$$-H(\bar{\mu}_w) \leq -\sum_{\ell=1}^m \lambda_\ell H(\mu_\ell) - \kappa \sum_{\ell \neq k} \lambda_\ell \lambda_k W^2_2(\mu_\ell, \mu_k)$$

and equivalently:

$$H(\bar{\mu}_w) \geq \sum_{\ell=1}^m \lambda_\ell H(\mu_\ell) + \kappa \sum_{\ell \neq k} \lambda_\ell \lambda_k W^2_2(\mu_\ell, \mu_k).$$

Note that this statement is true for a weaker notion of convexity that is displacement convexity and $\kappa$ depends on Ricci curvature.

**Lemma 2** (Entropic Regularized W. Barycenters: Controllable entropy via $\varepsilon$). Assume that $\mu_\ell$ are such that $\mu_{\ell,i} > 0$, for $\ell = 1 \ldots m$, $i = 1 \ldots M$, we have:

$$H(\bar{\mu}_w) + H(\mu_\ell) \geq -\sum_{j=1}^M \sum_{i=1}^M \gamma_{ij}^\ell \log(\gamma_{ij}^\ell), \forall \ell = 1 \ldots m.$$
Proof. Let $\gamma \in \mathbb{R}^{N \times M}_+$ be a coupling between $p \in \Delta_M$ and $q \in \Delta_N, q_j > 0$ we have: $\gamma^T 1_N = p$ and $\gamma 1_M = q$, we have:

$$p_j = \sum_{i=1}^N \gamma_{ij} = \sum_{i=1}^N q_i \frac{\gamma_{ij}}{q_i}, \ j = 1 \ldots M$$

Let $a_i = \left( \frac{\gamma_{ij}}{q_i} \right)_{j=1}^{M}$. Note that $\sum_{j=1}^M \frac{\gamma_{ij}}{q_i} = \frac{q_i}{q_i} = 1$, hence $a_i \in \Delta_M$. Hence $p$ can be written as a convex combination of probabilities:

$$p = \sum_{i=1}^N q_i a_i$$

Now the entropy of the convex combination is higher than convex combination of entropies (the entropy is concave):

$$H(p) \geq \sum_{i=1}^N q_i H(a_i)$$

$$H(a_i) = -\sum_{j=1}^M \frac{\gamma_{ij}}{q_i} \log \left( \frac{\gamma_{ij}}{q_i} \right)$$

$$= -\frac{1}{q_i} \left( \sum_{j=1}^M \gamma_{ij} \log(\gamma_{ij}) - \log(q_i) \sum_{i=1}^M \gamma_{ij} \right)$$

$$= -\frac{1}{q_i} \left( \sum_{i=1}^M \gamma_{ij} \log(\gamma_{ij}) - \log(q_i)q_i \right)$$

Hence:

$$H(p) \geq -\sum_{j=1}^N \sum_{i=1}^M \gamma_{ij} \log(\gamma_{ij}) + \sum_{i=1}^N q_i \log(q_i) = -\sum_{j=1}^N \sum_{i=1}^M \gamma_{ij} \log(\gamma_{ij}) - H(q)$$

Hence:

$$H(p) + H(q) \geq -\sum_{i=1}^N \sum_{j=1}^M \gamma_{ij} \log(\gamma_{ij})$$