Unbalanced Optimal Transport: A Unified Framework for Object Detection

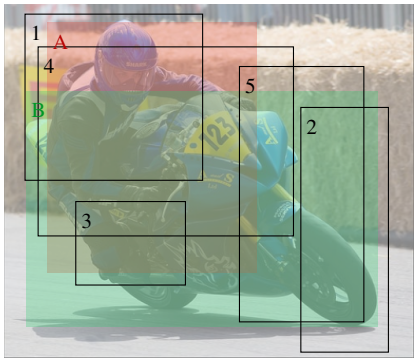
Henri De Plaen[1*] Pierre-François De Plaen[2*] Johan A. K. Suykens[1] Marc Proesmans[2]
Tinne Tuytelaars[2] Luc Van Gool[2,3]
[1]ESAT-STADIUS, KU Leuven, Belgium [2]ESAT-PSI, KU Leuven, Belgium
[3]Computer Vision Lab, ETH Zürich, Switzerland

Abstract

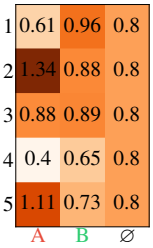
During training, supervised object detection tries to correctly match the predicted bounding boxes and associated classification scores to the ground truth. This is essential to determine which predictions are to be pushed towards which solutions, or to be discarded. Popular matching strategies include matching to the closest ground truth box (mostly used in combination with anchors), or matching via the Hungarian algorithm (mostly used in anchor-free methods). Each of these strategies comes with its own properties, underlying losses, and heuristics. We show how Unbalanced Optimal Transport unifies these different approaches and opens a whole continuum of methods in between. This allows for a finer selection of the desired properties. Experimentally, we show that training an object detection model with Unbalanced Optimal Transport is able to reach the state-of-the-art both in terms of Average Precision and Average Recall as well as to provide a faster initial convergence. The approach is well suited for GPU implementation, which proves to be an advantage for large-scale models.

1. Introduction

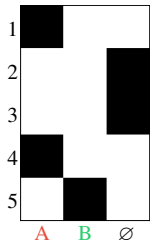
Object detection models are in essence multi-task models, having to both localize objects in an image and classify them. In the context of supervised learning, each of these tasks heavily depends on a matching strategy. Indeed, determining which predicted object matches which ground truth object is a non-trivial yet essential task during the training (Figure 1a). In particular, the matching strategy must ensure that there is ideally exactly one prediction per ground truth object, at least during inference. Various strategies have emerged, often relying on hand-crafted components. They are proposed as scattered approaches that seem to have nothing in common, at least at first glance.



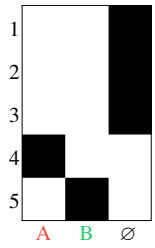
(a) Image №163 from the COCO training dataset. The ground truth boxes are colored, and the predictions are outlined in black.



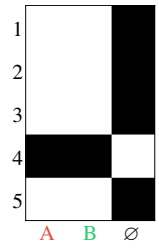
(b) Costs between the predictions and the ground truth ($1 - \text{GIoU}$). The background cost is $c_\varnothing = 0.8$.



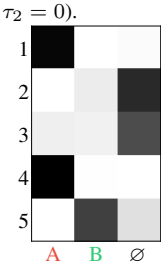
(c) Prediction to best ground truth (Unbalanced OT with $\epsilon = 0$, $\tau_1 \to +\infty$ and $\tau_2 = 0$).



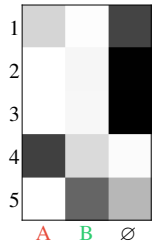
(d) Hungarian matching (OT with $\epsilon = 0$, $\tau_1 \to +\infty$ and $\tau_2 \to +\infty$).



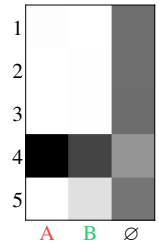
(e) Ground truth to best prediction (Unbalanced OT with $\epsilon = 0$, $\tau_1 = 0$ and $\tau_2 \to +\infty$).



(f) Unbalanced OT with $\epsilon = 0.05$, $\tau_1 = 100$ and $\tau_2 = 0.01$.



(g) OT with $\epsilon = 0.05$ ($\tau_1 \to +\infty$ and $\tau_2 \to +\infty$).



(h) Unbalanced OT with $\epsilon = 0.05$, $\tau_1 = 0.01$ and $\tau_2 = 100$.

Figure 1. Different matching strategies. All are particular cases of *Unbalanced Optimal Transport*.

*These authors contributed equally.

1.1. A Unifying Framework

To perform any match, a matching cost has to be determined. The example at Fig. 1b uses the *Generalized Intersection over Union* (GIoU) [46]. Given such a cost matrix, matching strategies include:

- Matching each prediction to the closest ground truth object. This often requires that the cost lies under a certain threshold [37, 45, 44, 33], to avoid matching predictions that may be totally irrelevant for the current image. The disadvantage of this strategy is its redundancy: many predictions may point towards the same ground truth object. In Fig. 1c, both predictions 1 and 4 are matched towards ground truth object A. Furthermore, some ground truth objects may be unmatched. A solution to this is to increase the number of predicted boxes drastically. This is typically the case with anchors boxes and region proposal methods.

- The opposite strategy is to match each ground truth object to the best prediction [25, 37]. This ensures that there is no redundancy and every ground truth object is matched. This also comes with the opposite problem: multiple ground truth objects may be matched to the same prediction. In Fig. 1e, both ground truth objects A and B are matched to prediction 4. This can be mitigated by having more predictions, but then many of those are left unmatched, slowing convergence [37].

- A compromise is to perform a *Bipartite Matching* (BM), using the *Hungarian algorithm* [29, 40], for example [6, 55]. The matching is one-to-one, minimizing the total cost (Definition 2). Every ground truth object is matched to a unique prediction, thus reducing the number of predictions needed, as shown in Fig. 1d. A downside is that the one-to-one matches may vary from one epoch to the next, again slowing down convergence [31]. This strategy is difficult to parallelize, *i.e.* to take advantage of GPU architectures.

All of these strategies have different properties and it seems that one must choose either one or the other, optionally combining them using savant heuristics [37]. There is a need for a unifying framework. As we show in this paper, *Unbalanced Optimal Transport* [9] offers a good candidate for this (Figure 1). It not only unifies the different strategies here above, but also allows to explore all cases in between. The cases presented in Figures 1c, 1d and 1e correspond to the limit cases. This opens the door for all intermediate settings. Furthermore, we show how regularizing the problem induces smoother matches, leading to faster convergence of DETR, avoiding the problem described for the BM. In addition, the particular choice of entropic regularization leads to a class of fast parallelizable algorithms on GPU known as *scaling algorithms* [10, 8], of which we provide a compiled implementation on GPU. Our code and additional resources are publicly available[1].

1.2. Related Work

Matching Strategies Most *two-stage* models often rely on a huge number of initial predictions, which is then progressively reduced in the region proposal stage and refined in the classification stage. Many different strategies have been proposed for the initial propositions and subsequent reductions, ranging from training no deep learning networks [21], to only train those for the propositions [20, 32, 25], to training networks for both propositions and reductions [45, 42, 24, 5, 11]. Whenever a deep learning network is trained, each prediction is matched to the closest ground truth object provided it lies beneath a certain threshold. Moreover, the final performance of these models heavily depends on the hand-crafted anchors [35].

Many *one-stage* models rely again on predicting a large number of initial predictions or *anchor boxes*, covering the entire image. As before, each anchor box is matched towards the closest ground truth object with certain threshold constraints [44, 33]. In [37], this is combined with matching each ground truth object to the closest anchor box and a specific ratio heuristic between the matched and unmatched predictions. The matching of the fixed anchors is justified to avoid a collapse of the predictions towards the same ground truth objects. Additionally, this only works if the number of initial predictions is sufficiently large to ensure that every ground truth object is matched by at least one prediction. Therefore, it requires further heuristics, such as *Non-Maximal Suppression* (NMS) to guarantee a unique prediction per ground truth object, at least during the inference.

By using the *Hungarian algorithm*, DETR [6] removed the need for a high number of initial predictions. The matched predictions are improved with a multi-task loss, and the remaining predictions are trained to predict the background class $\varnothing$. Yet, the model converges slowly due to the instability of BM, causing inconsistent optimization goals at early training stages [31]. Moreover, the sequential nature of the Hungarian algorithm does not take full advantage of the GPU architecture. Several subsequent works accelerate the convergence of DETR by improving the architecture of the model [55, 36] and by adding auxiliary losses [31], but not by exploring the matching procedure.

Optimal Transport The theory of *Optimal Transport* (OT) emerges from an old problem [38], relaxed by a newer formulation [26]. It gained interest in the machine learning community since the re-discovery of *Sinkhorn's algorithm* [10] and opened the door for improvements in a wide variety of applications ranging from graphical models [39], kernel methods [28, 13], loss design [17],

[1] https://hdeplaen.github.io/uotod

auto-encoders [50, 27, 47] or generative adversarial networks [3, 22].

More recent incursions in computer vision have been attempted, *e.g.* for the matching of predicted classes [23], a loss for rotated object boxes [54] or a new metric for performance evaluation [41]. Considering the matching of predictions to ground truth objects, recent attempts using OT bare promising results [18, 19]. However, when the *Hungarian algorithm* is mentioned, it is systematically presented in opposition to OT [18, 53]. We lay a rigorous connection between those two approaches in computer vision.

Unbalanced OT has seen a much more recent theoretical development [9, 7]. The hard mass conservation constraints in the objective function are replaced by soft penalization terms. Its applications are scarcer, but we must mention here relatively recent machine learning applications in motion tracking [30] and domain adaptation [16].

1.3. Contributions

1. We propose a unifying matching framework based on *Unbalanced Optimal Transport*. It encompasses both the *Hungarian algorithm*, the matching of the predictions to the closest ground truth boxes and the ground truth boxes to the closest predictions;

2. We show that these three strategies correspond to particular limit cases and we subsequently present a much broader class of strategies with varying properties;

3. We demonstrate how entropic regularization can speed up the convergence during training and additionally take advantage of GPU architectures;

4. We justify the relevancy of our framework by exploring its interaction with NMS and illustrate how it is on par with the state-of-the-art.

1.4. Notations and Definitions

Notations Throughout the paper, we use small bold letters to denote a vector $\boldsymbol{a} \in \mathbb{R}^N$, with elements $a_i \in \mathbb{R}$. Similarly, matrices are denoted by bold capital letters such as $\boldsymbol{A} \in \mathbb{R}^{N \times M}$, with elements $A_{i,j} \in \mathbb{R}$. The notation $\mathbf{1}_N$ represents a column-vector of ones, of size N, and $\mathbf{1}_{N \times M}$ the matrix equivalent of size $N \times M$. The identity matrix of size N is $\boldsymbol{I}_{N,N}$. With $[\![N]\!] = \{1, 2, \dots, N\}$, we denote the set of integers from 1 to N. The probability simplex uses the notation $\Delta^N = \left\{ \boldsymbol{u} \in \mathbb{R}^N_{\geq 0} \middle| \sum_i u_i = 1 \right\}$ and represents the set of discrete probability distributions of dimension N. This extends to the set of discrete joint probability distributions $\Delta^{N \times M}$.

Definitions For each image, the set $\{\hat{\boldsymbol{y}}_i\}_{i=1}^{N_p}$ denotes the predictions and $\{\boldsymbol{y}_j\}_{j=1}^{N_g}$ the ground truth samples. Each ground truth sample combines a target class and a bounding box position: $\boldsymbol{y}_j = [\, \boldsymbol{c}_j,\, \boldsymbol{b}_j \,] \in \mathbb{R}^{N_c+4}$ where $\boldsymbol{c}_j \in \{0,1\}^{N_c}$ is the target class in one-hot encoding with N_c the number of classes and $\boldsymbol{b}_j \in [0,1]^4$ defines the relative bounding box center coordinates and dimensions. The predictions are defined similarly $\hat{\boldsymbol{y}}_i = [\, \hat{\boldsymbol{c}}_i,\, \hat{\boldsymbol{b}}_i \,] \in \mathbb{R}^{N_c+4}$, but the predicted classes may be non-binary $\hat{\boldsymbol{c}}_i \in [0,1]^{N_c}$. Sometimes, predictions are defined relatively to fixed anchor boxes $\tilde{\boldsymbol{b}}_i$.

2. Optimal Transport

In this section, we show how *Optimal Transport* and then its *Unbalanced* extension unify both the *Hungarian algorithm* used in DETR [6], and matching each prediction to the closest ground truth object used in both Faster R-CNN [45] and SSD [37]. We furthermore stress the advantages of entropic regularization, both computationally and qualitatively. This allows us to explore a new continuum of matching methods, with varying properties.

Definition 1 (Optimal Transport)**.** *Given a distribution $\boldsymbol{\alpha} \in \Delta^{N_p}$ associated to the predictions $\{\hat{\boldsymbol{y}}_i\}_{i=1}^{N_p}$, and another distribution $\boldsymbol{\beta} \in \Delta^{N_g}$ associated with the ground truth objects $\{\boldsymbol{y}_j\}_{j=1}^{N_g}$. Let us consider a pair-wise matching cost $\mathcal{L}_{match}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j)$ between a prediction $\hat{\boldsymbol{y}}_i$ and a ground truth object $\boldsymbol{y}_j$. We now define* Optimal Transport (OT) *as finding the match $\boldsymbol{P}$ that minimizes the following problem:*

$$\hat{\boldsymbol{P}} = \underset{\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\arg\min} \left\{ \sum_{i,j=1}^{N_p, N_g} P_{i,j} \mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right) \right\}, \quad (1)$$

with transport polytope (admissible solutions) $\mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \left\{ \boldsymbol{P} \in \mathbb{R}^{N_p \times N_g}_{\geq 0} : \sum_{j=1}^{N_g} P_{i,j} = \alpha_i, \sum_{i=1}^{N_p} P_{i,j} = \beta_j \right\}$.

Provided that certain conditions apply to the underlying cost $\mathcal{L}_{\text{match}}$, the minimum defines a distance between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, referred to as the Wasserstein distance $\mathcal{W}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ (for more information, we refer to monographs [52, 48, 43]; see also Appendix A.2).

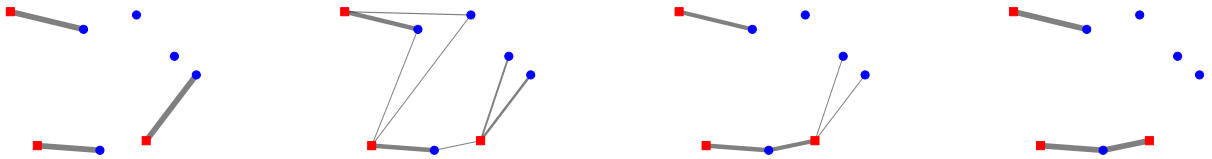
2.1. The Hungarian Algorithm

The Hungarian algorithm solves the *Bipartite Matching* (BM). We will now show how this is a particular case of Optimal Transport.

Definition 2 (Bipartite Matching)**.** *Given the same objects as in Definition 1, the* Bipartite Matching (BM) *minimizes the cost of the pairwise matches between the ground truth objects with the predictions:*

$$\hat{\sigma} = \arg\min \left\{ \sum_{j=1}^{N_g} \mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_{\sigma(j)}, \boldsymbol{y}_j\right) : \sigma \in \mathcal{P}_{N_g}([\![N_p]\!]) \right\}, \quad (2)$$

where $\mathcal{P}_{N_g}([\![N_p]\!]) = \left\{ \sigma \in \mathcal{P}([\![N_p]\!]) \,\middle|\, |\sigma| = N_g \right\}$ is the set of possible combinations of N_g in N_p, with $\mathcal{P}([\![N_p]\!])$ the power set of $[\![N_p]\!]$ (the set of all subsets).



(a) BM as a particular case of OT with no regularization ($\epsilon = 0$). The *Hungarian algorithm* obtains the same solution.

(b) OT with regularization ($\epsilon \neq 0$). The regularization smoothens the matching allowing for multiple connections.

(c) Unbalanced OT with regularization ($\epsilon \neq 0$ and $\tau_1 \ll \tau_2$). The smoothing is also visible.

(d) Matching each ground truth object to the closest prediction as Unbalanced OT without regularization with $\epsilon = 0$, $\tau_1 = 0$ and $\tau_2 \to \infty$.

Figure 2. Example of the influence of the parameters. The blue dots represent predictions $\hat{\boldsymbol{y}}_i$. The red squares represent ground truth objects $\boldsymbol{y}_j$. The distributions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are defined as in Prop. 1. The thickness of the lines is proportional to the amount transported $P_{i,j}$. Only sufficiently thick lines are plotted. The dummy *background* ground truth $\boldsymbol{y}_{N_g+1} = \varnothing$ is not shown, nor are the connections to it.

BM tries to assign each ground truth $\boldsymbol{y}_j$ to a different prediction $\hat{\boldsymbol{y}}_i$ in a way to minimize the total cost. In contrast to OT, BM does not consider any underlying distributions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, all ground truth objects and predictions are implicitly considered to be of same mass. Furthermore, it only allows one ground truth to be matched to a unique prediction, some of these predictions being left aside and matched to nothing (which is then treated as a matching to the background $\varnothing$). The OT must match all ground truth objects to all predictions, not allowing any predictions to be left aside. However, the masses of the ground truth objects are allowed to be split between different predictions and inversely, as long as their masses correctly sum up ($\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})$).

Particular Case of OT A solution for an imbalanced number of predictions compared to the number of ground truth objects would be to add dummy ground truth objects—the background $\varnothing$—to even the balance. Concretely, one could add a new ground truth $\boldsymbol{y}_{N_g+1} = \varnothing$, with the mass equal to the unmatched number of predictions. In fact, doing so directly results in performing a BM.

Proposition 1. *The Hungarian algorithm with N_p predictions and $N_g \leq N_p$ ground truth objects is a particular case of OT with $\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \subset \mathbb{R}^{N_p \times (N_g+1)}$, consisting of the predictions and the ground truth objects, with the background added $\{\boldsymbol{y}_j\}_{j=1}^{N_g+1} = \{\boldsymbol{y}_j\}_{j=1}^{N_g} \cup (\boldsymbol{y}_{N_g+1} = \varnothing)$. The chosen underlying distributions are*

$$\boldsymbol{\alpha} = \frac{1}{N_p}[\underbrace{1,\ 1,\ 1,\ \ldots,\ 1}_{N_p \text{ predictions}}], \tag{3}$$

$$\boldsymbol{\beta} = \frac{1}{N_p}[\underbrace{1,\ 1,\ \ldots,\ 1}_{N_g \text{ ground truth objects}},\ \underbrace{(N_p - N_g)}_{\text{background } \varnothing}], \tag{4}$$

provided the background cost is constant: $\mathcal{L}_{match}(\hat{\boldsymbol{y}}_i, \varnothing) = c_\varnothing$. In particular for $j \in [\![N_g]\!]$, we have $\hat{\sigma}(j) = \{i : P_{i,j} \neq 0\}$, or equivalently $\hat{\sigma}(j) = \{i : P_{i,j} = 1/N_p\}$.

Proof. We refer to Appendix B.1. □

In other words, we can read the matching to each ground truth in the columns of $\hat{\boldsymbol{P}}$. The last columns represents all the predictions matched to the background $\hat{\sigma}(N_g + 1)$. Alternatively and equivalently, we can read the matching of each prediction i in the rows, the ones being matched to the background have a $\hat{P}_{i,N_g+1} = 1/N_p$.

Solving the Problem Both OT and BM are linear programs. Using generic formulations would lead to a $(N_p + N_g + 1) \times N_p\,(N_g + 1)$ equality constraint matrix. It is thus better to exploit the particular bipartite structure of the problem. In particular, two families of algorithms have emerged: *Dual Ascent Methods* and *Auction Algorithms* [43]. The Hungarian algorithm is a particular case of the former and classically runs with an $\mathcal{O}\left(N_p^4\right)$ complexity [40], further reduced to cubic by [14]. Although multiple GPU implementations of a BM solver have been proposed [51, 12, 15], the problem remains poorly parallelizable because of its sequential nature. To allow for efficient parallelization, we must consider a slightly amended problem.

2.2. Regularization

We show here how we can replace the *Hungarian algorithm* by a class of algorithms well-suited for parallelization, obtained by adding an entropy regularization.

Definition 3 (OT with regularization)**.** *We consider a regularization parameter $\epsilon \in \mathbb{R}_{\geq 0}$. Extending Definition 1 (OT), we define the* Optimal Transport with regularization *as the following minimization problem:*

$$\hat{\boldsymbol{P}} = \underset{\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\arg\min} \left\{ \sum_{i,j=1}^{N_p, N_g} P_{i,j} \mathcal{L}_{match}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j) - \epsilon\, \mathrm{H}(\boldsymbol{P}) \right\}, \tag{5}$$

with $\mathrm{H} : \Delta^{N \times M} \to \mathbb{R}_{\geq 0} : \boldsymbol{P} \mapsto -\sum_{i,j} P_{i,j}(\log(P_{i,j}) - 1)$ the entropy of the match $\boldsymbol{P}$, with $0 \ln(0) = 0$ by definition.

Sinkhorn's Algorithm The entropic regularization used when finding the match $\hat{\boldsymbol{P}}$ ensures that the problem is

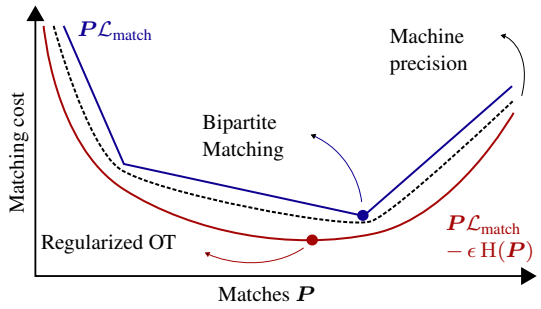


Figure 3. Effect of the regularization on the minimization of the matching cost. The red line corresponds to the regularized problem ($\epsilon \neq 0$) and the blue to the unregularized one ($\epsilon = 0$).

smooth for $\epsilon \neq 0$ (see Figure 3). The advantage is that it can now be solved very efficiently using *scaling algorithms* and in this particular case the algorithm of *Sinkhorn*. It is particularly suited for parallelization [10], with some later speed refinements [2, 1]. Reducing the regularization progressively renders the scaling algorithms numerically unstable, although some approaches have been proposed to reduce the regularization further by working in log-space [49, 8]. In the limit of $\epsilon \to 0$, we recover the exact OT (Definition 1) and the scaling algorithms cannot be used anymore. Parallelization is lost and we must resolve to use the sequential algorithms developed in Section 2.1. In brief, regularization allows to exploit GPU architectures efficiently, whereas the Hungarian algorithm and similar cannot.

Smoother Matches When no regularization is used as in the Hungarian algorithm, close predictions and ground truth objects can exchange their matches from one epoch to the other, during the training. This causes a slow convergence of DETR in the early stages of the training [31]. The advantage of the regularization not only lies in the existence of efficient algorithms but also allows for a reduction of sparsity. This results in a less drastic match than the Hungarian algorithm obtains. A single ground truth could be matched to multiple predictions and inversely. The proportion of these multiple matches is controlled by the regularization parameter ϵ. An illustration can be found in Figures 2a and 2b.

2.3. Unbalanced Optimal Transport

We will now show how considering soft constraints instead of hard leads to an even greater generalization of the various matching techniques used in object detection models. In particular, matching each prediction to the closest ground truth is a limit case of the *Unbalanced OT*.

Definition 4 (Unbalanced OT)**.** *We consider two constraint parameters $\tau_1, \tau_2 \in \mathbb{R}_{\geq 0}$. Extending Definition 3 (OT with regularization), we define the* Unbalanced OT with regularization *[8] as the following minimization problem:*

$$\hat{\boldsymbol{P}} = \underset{\boldsymbol{P} \in \mathbb{R}_{\geq 0}^{N_p \times N_g}}{\arg\min} \Big\{ \epsilon\,\mathrm{KL}(\boldsymbol{P} \,\|\, \boldsymbol{K}_\epsilon) + \tau_1 \mathrm{KL}(\boldsymbol{P}\mathbf{1}_{N_g} \,\|\, \boldsymbol{\alpha}) + \tau_2 \mathrm{KL}(\mathbf{1}_{N_p}^\top \boldsymbol{P} \,\|\, \boldsymbol{\beta}) \Big\}, \tag{6}$$

where $\mathrm{KL} : \mathbb{R}_{\geq 0}^{N \times M} \times \mathbb{R}_{>0}^{N \times M} \to \mathbb{R}_{\geq 0} : (\boldsymbol{U}, \boldsymbol{V}) \mapsto \sum_{i,j=1}^{N \times M} U_{i,j} \log(U_{i,j}/V_{i,j}) - U_{i,j} + V_{i,j}$ *is the* Kullback-Leibler divergence *– also called* relative entropy *– between matrices or vectors when $M = 1$, with $0 \ln(0) = 0$ by definition. The Gibbs kernel $\boldsymbol{K}_\epsilon$ is given by $(K_\epsilon)_{i,j} = \exp\left(-\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)/\epsilon\right)$.*

We can see by development that the first term corresponds to the matching term $\boldsymbol{P}\mathcal{L}_{\text{match}}$ and an extension of the entropic regularization term $\mathrm{H}(\boldsymbol{P})$. The two additional terms replace the transport polytope's hard constraints $\mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ that required an exact equality of mass for both marginals. These new soft constraints allow for a more subtle sensitivity to the mass constraints as it allows to slightly diverge from them. It is clear that in the limit of $\tau_1, \tau_2 \to +\infty$, we recover the "balanced" problem (Definition 3). This definition naturally also defines Unbalanced OT without regularization if $\epsilon = 0$. The matching term would remain and the entropic one disappear.

Matching to the Closest Another limit case is however particularly interesting in the quest for a unifying framework of the matching strategies. If the mass constraint is to be perfectly respected for the predictions ($\tau_1 \to \infty$), but not at all for the ground truth objects ($\tau_2 = 0$), it suffices to assign the closest ground truth to each prediction. The same ground truth object could be assigned to multiple predictions and another could not be matched at all, not respecting the hard constraint for the ground truth $\boldsymbol{\beta}$. Each prediction however is exactly assigned once, perfectly respecting the mass constraint for the predictions $\boldsymbol{\alpha}$. By assigning a low enough value to the background, a prediction would be assigned to it provided all the other ground truth objects are further. In other words, the background cost would play the role of a *threshold* value.

Proposition 2 (Matching to the closest)**.** *We consider the same objects as Proposition 1. In the limit of $\tau_1 \to \infty$ and $\tau_2 = 0$, Unbalanced OT (Definition 4) without regularization ($\epsilon = 0$) admits as solution each prediction being matched to the closest ground truth object unless that distance is greater than a threshold value $\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_{N_g+1} = \varnothing\right) = c_\varnothing$. It is then matched to the background $\varnothing$. In particular, we have*

$$\hat{P}_{i,j} = \begin{cases} \frac{1}{N_p} & \text{if } j = \arg\min_{j \in [\![N_g+1]\!]} \left\{\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right\}, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Proof. We refer to Appendix B.2. □

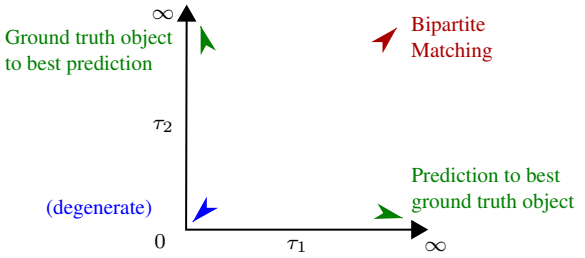


Figure 4. Limit cases of Unbalanced OT without regularization ($\epsilon = 0$).

The converse also holds. If the ground truth objects mass constraints were to be perfectly respected ($\tau_2 \to \infty$), but not the predictions ($\tau_1 \to 0$), each ground truth would then be matched to the closest prediction. The background would be matched to the remaining predictions. Some predictions could not be matched and other ones multiple times. The limits of Unbalanced OT are illustrated in Fig. 4. By setting the threshold sufficiently high, we get an exact minimum, i.e., where every prediction is matched to the closest ground truth. This can be observed in Figure 2d.

Scaling Algorithm Similarly as before, adding entropic regularization ($\epsilon \neq 0$) to the *Unbalanced OT* allows it to be solved efficiently on GPU with a scaling algorithm, as an extension of Sinkhorn's algorithm [8, 7]. The regularization still also allows for smoother matches, as shown in Figure 2c.

Softmax In the limit of $\tau_1 \to +\infty$ and $\tau_2 = 0$, the solution corresponds to a softmax over the ground truth objects for each prediction. The regularization ε controls then the "softness" of the softmax, with $\varepsilon = 1$ corresponding to the conventional softmax and $\varepsilon \to 0$ the matching to the closest. We refer to Appendix C.2 for more information.

3. Matching

Following previous work [6, 55, 45, 44, 37], we define a multi-task matching cost between a prediction $\hat{\boldsymbol{y}}_i$ and a ground truth object $\boldsymbol{y}_j$ as the composition of a classification loss ensuring that similar object classes are matched together and a localization loss ensuring the correspondence of the positions and shapes of the matched boxes $\mathcal{L}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j) = \mathcal{L}_{\text{classification}}(\hat{\boldsymbol{c}}_i, \boldsymbol{c}_j) + \mathcal{L}_{\text{localization}}(\hat{\boldsymbol{b}}_i, \boldsymbol{b}_j)$. Most models, however, do not use the same loss to determine the matches as the one used to train the model. We therefore refer to these two losses as $\mathcal{L}_{\text{match}}$ and $\mathcal{L}_{\text{train}}$. The training procedure is the following: first find a match $\hat{\boldsymbol{P}}$ given a matching strategy and matching cost $\mathcal{L}_{\text{match}}$, then compute the loss $N_p \sum_{i=1}^{N_p} \sum_{j=1}^{N_g} \hat{P}_{ij} \mathcal{L}_{\text{train}}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j)$ where the particular training loss for the background ground truth includes only a classification term $\mathcal{L}_{\text{train}}(\hat{\boldsymbol{y}}_i, \varnothing) = \mathcal{L}_{\text{classification}}(\hat{\boldsymbol{c}}_i, \varnothing)$.

3.1. Detection Transformer (DETR)

The object detection is performed by matching the predictions to the ground truth boxes with the *Hungarian algorithm* applied to the loss $\mathcal{L}_{\text{match}}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j) = \lambda_{\text{prob}}(1 - \langle \hat{\boldsymbol{c}}_i, \boldsymbol{c}_j \rangle) + \lambda_{\ell^1} \|\hat{\boldsymbol{b}}_i - \boldsymbol{b}_j\|_1 + \lambda_{\text{GIoU}}(1 - \text{GIoU}(\hat{\boldsymbol{b}}_i, \boldsymbol{b}_j))$ (Definition 2). To do so, the number of predictions and ground truth boxes must be of the same size. This is achieved by padding the ground truths with $(N_p - N_g)$ dummy *background* $\varnothing$ objects. Essentially, this is the same as what is developed in Proposition 1. The obtained match is then used to define an object-specific loss, where each matched prediction is pushed toward its corresponding ground truth object. The predictions that are not matched to a ground truth object are considered to be matched with the background and are pushed to predict the background class. The training loss uses the cross-entropy (CE) for classification: $\mathcal{L}_{\text{train}}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j) = \lambda_{\text{CE}} \mathcal{L}_{\text{CE}}(\hat{\boldsymbol{c}}_i, \boldsymbol{c}_j) + \lambda_{\ell^1} \|\hat{\boldsymbol{b}}_i - \boldsymbol{b}_j\|_1 + \lambda_{\text{GIoU}}(1 - \text{GIoU}(\hat{\boldsymbol{b}}_i, \boldsymbol{b}_j))$. By directly applying Proposition 1 and adding entropic regularization (Definition 3), we can use *Sinkhorn's algorithm* and push each prediction $\hat{\boldsymbol{y}}_i$ to ground truth $\boldsymbol{y}_j$ according to weight $\hat{P}_{i,j}$. In particular, for any non-zero $\hat{P}_{i,N_g+1} \neq 0$, the prediction $\hat{\boldsymbol{y}}_i$ is pushed toward the background $\boldsymbol{y}_{N_g+1} = \varnothing$ with weight $\hat{P}_{i,N_g+1}$.

3.2. Single Shot MultiBox Detector (SSD)

The Single Shot MultiBox Detector [37] uses a matching cost only comprised of the IoU between the fixed anchor boxes $\tilde{\boldsymbol{b}}_i$ and the ground truth boxes: $\mathcal{L}_{\text{match}}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j) = 1 - \text{IoU}(\tilde{\boldsymbol{b}}_i, \boldsymbol{b}_j)$ (the GIoU was not published yet [46]). Each ground truth is first matched toward the closest anchor box. Anchor boxes are then matched to a ground truth object if the matching cost is below a threshold of 0.5. In our framework, this corresponds to applying $\tau_1 = 0$ and $\tau_2 \to \infty$ for the first phase and then $\tau_1 \to \infty$ and $\tau_2 = 0$ with $c_\varnothing = 0.5$ (see Proposition 2). Here again, by adding entropic regularization (Definition 4), we can solve this using a *scaling algorithm*. We furthermore can play with the parameters τ_1 and τ_2 to make the matching tend slightly more towards a matching done with the *Hungarian algorithm* (Figure 2). Again, the training uses a different loss than the matching, in particular $\mathcal{L}_{\text{train}}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j) = \lambda_{\text{CE}} \mathcal{L}_{\text{CE}}(\hat{\boldsymbol{c}}_i, \boldsymbol{c}_j) + \lambda_{\text{smooth } \ell^1} \mathcal{L}_{\text{smooth } \ell^1}(\hat{\boldsymbol{b}}_i, \boldsymbol{b}_j)$.

Hard Negative Mining Instead of using all negative examples $N_{\text{neg}} = (N_p - N_g)$ (predictions matched to background), the method sorts them using the highest confidence loss $\mathcal{L}_{\text{CE}}(\hat{\boldsymbol{c}}_i, \varnothing)$ and picks the top ones so that the ratio between the hard negatives and positives $N_{\text{pos}} = N_g$ is at most 3 to 1. Since $\hat{\boldsymbol{P}}$ is non-binary, we define the number of negatives and positives to be the sum of the matches to the background $N_{\text{neg}} = N_p \sum_{i=1}^{N_p} \hat{P}_{i,(N_g+1)}$ and to the ground truth objects $N_{\text{pos}} = N_p \sum_{j=1}^{N_g} \sum_{i=1}^{N_p} \hat{P}_{ij}$. We verify that for any

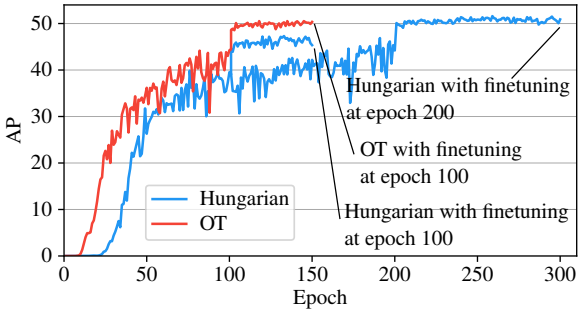


Figure 5. Convergence curves for DETR on the Color Boxes dataset. The model converges faster with a regularized matching.

	Model	Matching	τ_2	Epochs	AP	AR
Color Boxes	DETR	Hungarian	(∞)	300	**50.9**	**65.7**
	DETR	Hungarian	(∞)	150	45.3	60.7
	DETR	OT	(∞)	**150**	50.3	**65.7**
	D. DETR	Hungarian	(∞)	50	**64.0**	75.9
	D. DETR	OT	(∞)	50	63.5	**76.5**
COCO	D. DETR	Hungarian	(∞)	50	**44.5**	**63.0**
	D. DETR	OT	(∞)	50	44.2	62.0
	SSD300	Two Stage	—-	120	**24.9**	**36.8**
	SSD300	Unb. OT	0.01	120	24.7	36.4

Table 1. Object detection metrics for different models and loss functions on the Color Boxes and COCO datasets.

$\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})$, we have the same number of positives and negatives as the initial model: $N_{\text{neg}} = (N_p - N_g)$ and $N_{\text{pos}} = N_g$. Hence, hard negatives are the K predictions with the highest confidence loss $\hat{P}_{k,(N_g+1)}\mathcal{L}_{\text{CE}}(\hat{\boldsymbol{c}}_k, \varnothing)$ such that the mass of kept negatives is at most triple the number of positives: $N_p \sum_{k=1}^{K} \hat{P}^s_{k,(N_g+1)} \leq 3N_{\text{pos}}$, where $\hat{\boldsymbol{P}}^s$ is a permutation of transport matrix $\hat{\boldsymbol{P}}$ with rows sorted by highest confidence loss.

4. Experimental Results & Discussion

We show that matching based on *Unbalanced Optimal Transport* generalizes many different matching strategies and performs on par with methods that use either *Bipartite Matching* or anchor boxes along with matching each prediction to the closest ground truth box with a threshold. We then analyze the influence of constraint parameter τ_2 by training SSD with and without NMS for multiple parameter values. Finally, we show that OT with entropic regularization both improves the convergence and is faster to compute than the Hungarian algorithm in case of many matches.

4.1. Setup

Datasets We perform experiments on a synthetic object detection dataset with 4.800 training and 960 validation images and on the large-scale COCO [34] dataset with 118,287 training and 5,000 validation test images. We report on mean Average Precision (AP) and mean Average Recall (AR). The two metrics are an average of the per-class metrics following COCO's official evaluation procedure. For the Color Boxes synthetic dataset, we uniformly randomly draw between 0 and 30 rectangles of 20 different colors from each image. Appendix I provides the detailed generation procedure and sample images.

Training For a fair comparison, the classification and localization costs for matching and training are identical to the ones used by the models. Unless stated otherwise, we train the models with their default hyper-parameter sets. DETR and Deformable DETR are trained with hyper-parameters $\lambda_{\text{prob}} = \lambda_{\text{CE}} = 2$, $\lambda_{\ell^1} = 5$ and $\lambda_{\text{GIoU}} = 2$. For Deformable DETR, we found the classification cost to be overwhelmed by the localization costs in the regularized minimization problem (Definition 3). We therefore set $\lambda_{\text{prob}} = 5$. We, however keep $\lambda_{\text{CE}} = 2$ so that the final loss value for a given matching remains unchanged. SSD is trained with original hyper-parameters $\lambda_{\text{CE}} = \lambda_{\text{smooth }\ell^1} = 1$. For OT, we set the entropic regularization to $\epsilon = \epsilon_0/(\log(2N_p) + 1)$ where $\epsilon_0 = 0.12$ for all models (App. D). In the following experiments, the Unbalanced OT is solved with multiple values of τ_2 whereas τ_1 is fixed to a large value $\tau_1 = 100$ to simulate a hard constraint. In practice, we limit the number of iterations of the scaling algorithm. This provides a good enough approximation [19].

4.2. Unified Matching Strategy

DETR and Deformable DETR Convergence curves for DETR on the Color Boxes dataset are shown in Fig. 5 and associated metrics are presented in Table 1. DETR converges in half the number of epochs with the regularized balanced OT formulation. This confirms that one reason for slow DETR convergence is the discrete nature of BM, which is unstable, especially in the early stages of training. Training the model for more epochs with either BM or OT does not improve metrics as the model starts to overfit. Appendix E provides qualitative examples and a more detailed convergence analysis. We evaluate how these results translate to faster converging DETR-like models by additionally training Deformable DETR [55]. In addition to model improvements, Deformable DETR makes three times more predictions than DETR and uses a sigmoid focal loss [33] instead of a softmax cross-entropy loss for both classification costs. Table 1 gives results on Color Boxes and COCO. We observe that the entropy term does not lead to faster convergence. Indeed, Deformable DETR converges in 50 epochs with both matching strategies. Nevertheless, both OT and bipartite matching lead to similar AP and AR.

SSD and the Constraint Parameter To better understand how unbalanced OT bridges the gap between DETR's

Matching	τ_2	with NMS		w/o NMS	
		AP	AR	AP	AR
Two Stage	—	**51.6**	**67.0**	23.2	**77.8**
Unb. OT	0.01	51.1	66.3	25.3	76.5
Unb. OT	0.1	50.9	66.8	35.9	75.4
Unb. OT	1	48.3	64.4	44.3	73.4
Unb. OT	10	48.0	64.1	44.9	72.9
OT	(∞)	48.1	64.3	**45.2**	73.0

Table 2. Comparison of matching strategies on the Color Boxes dataset. SSD300 is evaluated both with and without NMS.

and SSD's matching strategies, we analyze the variation in performance of SSD for different values of τ_2. Results for an initial learning rate of 0.0005 are displayed in Table 2. In the second row, the parameter value is close to zero. From Proposition 2 and when $\epsilon \to 0$, each prediction is matched to the closest ground truth box unless the matching cost exceeds 0.5. Thus, multiple predictions are matched to each ground truth box, and NMS is needed to eliminate near duplicates. When NMS is removed, AP drops by 25.8 points and AR increases by 10.2 points. We observe similar results for the original SSD matching strategy (1st row), which suggests matching each ground truth box to the closest anchor box does not play a huge role in the two-stage matching procedure from SSD. The lower part of Table 1 shows the same for COCO. When $\tau_2 \to +\infty$, one recovers the balanced formulation used in DETR (last row). Removing NMS leads to a 2.9 points drop for AP and a 9.7 points increase for AR. Depending on the field of application, it may be preferable to apply a matching strategy with a low τ_2 and with NMS when precision is more important or without NMS when the recall is more important. Moreover, varying parameter τ_2 offers more control on the matching strategy and therefore on the precision-recall trade-off [4].

Computation Time For a relatively small number of predictions, implementations of Sinkhorn perform on par with the Hungarian algorithm (Fig. 6). The "balanced" algorithm is on average 2.6ms slower than the Hungarian algorithm for 100 predictions (DETR) and 1.5ms faster for 300 predictions (Deformable DETR). For more predictions, GPU parallelization of the Sinkhorn algorithm makes a large difference (more than 50x speedup). As a reference point, SSD300 and SSD512 make $8,732$ and $24,564$ predictions.

5. Conclusion and Future Work

Throughout the paper, we showed both theoretically and experimentally how *Unbalanced Optimal Transport* unifies the *Hungarian algorithm*, matching each ground truth object to the best prediction and each prediction to the best ground truth, with or without threshold.

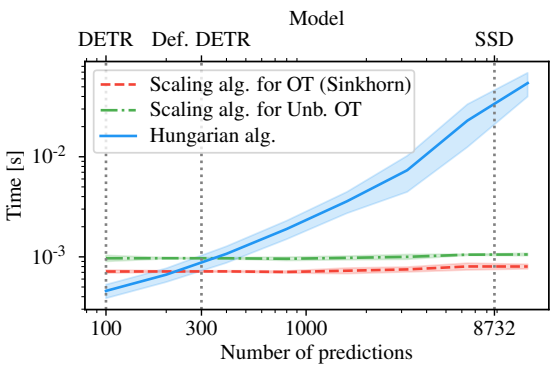


Figure 6. Average and standard deviation of the computation time for different matching strategies on COCO with batch size 16. The Hungarian algorithm is computed with *SciPy* and its time includes the transfer of the cost matrix from GPU memory to RAM. We run 20 Sinkhorn iterations. Computed with an Nvidia TITAN X GPU and Intel Core i7-4770K CPU @ 3.50GHz.

Experimentally, using OT and Unbalanced OT with entropic regularization is on par with the state-of-the-art for DETR, Deformable DETR and SSD. Moreover, we showed that entropic regularization lets DETR converge faster on the Color Boxes dataset and that parameter τ_2 offers better control of the precision-recall trade-off. Finally, we showed that the *scaling algorithms* compute large numbers of matches faster than the Hungarian algorithm.

Limitations and Future Work The convergence improvement of the regularized OT formulation compared to bipartite matching seems to hold only for DETR and on small-scale datasets. Further investigations may include Wasserstein-based matching costs for a further unification of the theory and the reduction of the entropy with time, as it seems to boost convergence only in early phases, but not in fine-tuning.

Acknowledgements

EU: The research leading to these results has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program / ERC Advanced Grant E-DUALITY (787960). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. Research Council KUL: Optimization frameworks for deep kernel machines C14/18/068. Flemish Government: FWO: projects: GOA4917N (Deep Restricted Kernel Machines: Methods and Foundations), PhD/Postdoc grant; This research received funding from the Flemish Government (AI Research Program). All the authors are also affiliated to Leuven.AI - KU Leuven institute for AI, B-3000, Leuven, Belgium.

References

[1] Mokhtar Z Alaya, Maxime Berar, Gilles Gasso, and Alain Rakotomamonjy. Screening sinkhorn algorithm for regularized optimal transport. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 5

[2] Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 5

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 3

[4] Michael Buckland and Fredric Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19, 1994. 8

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 2

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2, 3, 6

[7] Lenaic Chizat. *Unbalanced Optimal Transport : Models, Numerical Methods, Applications*. Theses, Université Paris sciences et lettres, Nov. 2017. 3, 6

[8] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018. 2, 5, 6

[9] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274(11):3090–3123, 2018. 2, 3

[10] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013. 2, 5

[11] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 29, 2016. 2

[12] Ketan Date and Rakesh Nagi. Gpu-accelerated hungarian algorithms for the linear assignment problem. *Parallel Computing*, 57:52–72, 2016. 4

[13] Henri De Plaen, Michaël Fanuel, and Johan AK Suykens. Wasserstein exponential kernels. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2020. 2

[14] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, apr 1972. 4

[15] Bas O Fagginger Auer and Rob H Bisseling. A gpu algorithm for greedy graph matching. In *Facing the Multicore-Challenge II*, pages 108–119. Springer, 2012. 4

[16] Kilian Fatras, Thibault Séjourné, Rémi Flamary, and Nicolas Courty. Unbalanced minibatch optimal transport; applications to domain adaptation. In *International Conference on Machine Learning*, pages 3186–3197. PMLR, 2021. 3

[17] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. *Advances in neural information processing systems*, 28, 2015. 2

[18] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 303–312, 2021. 3

[19] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 3, 7

[20] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, December 2015. 2

[21] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[22] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017. 3

[23] Yuzhuo Han, Xiaofeng Liu, Zhenfei Sheng, Yutao Ren, Xu Han, Jane You, Risheng Liu, and Zhongxuan Luo. Wasserstein loss-based deep object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 3

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 2

[26] L. Kantorovitch. On the translocation of masses. *Management Science*, 5(1):1–4, 1958. 2

[27] Soheil Kolouri, Phillip E Pope, Charles E Martin, and Gustavo K Rohde. Sliced wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. 3

[28] Soheil Kolouri, Yang Zou, and Gustavo K Rohde. Sliced wasserstein kernels for probability distributions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5258–5267, 2016. 2

[29] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 2

[30] John Lee, Nicholas P. Bertrand, and Christopher J. Rozell. Unbalanced optimal transport regularization for imaging problems. *IEEE Transactions on Computational Imaging*, 6:1219–1232, 2020. 3

[31] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 2, 5

[32] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[33] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 7

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 7

[35] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2):261–318, 2020. 2

[36] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. In *International Conference on Learning Representations*, 2021. 2

[37] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2, 3, 6

[38] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pages 666–704, 1781. 2

[39] Grégoire Montavon, Klaus-Robert Müller, and Marco Cuturi. Wasserstein training of restricted boltzmann machines. *Advances in Neural Information Processing Systems*, 29, 2016. 2

[40] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957. 2, 4

[41] Mayu Otani, Riku Togashi, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Shin'ichi Satoh. Optimal correction cost for object detection evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21107–21115, 2022. 3

[42] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 821–830, 2019. 2

[43] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 3, 4

[44] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2, 6

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 2, 3, 6

[46] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society. 2, 6

[47] Paul K Rubenstein, Bernhard Schoelkopf, and Ilya Tolstikhin. On the latent space of wasserstein auto-encoders. *arXiv preprint arXiv:1802.03761*, 2018. 3

[48] Filippo Santambrogio. Optimal transport for applied mathematicians. calculus of variations, pdes and modeling. 2015. 3

[49] Bernhard Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, 2019. 5

[50] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. 3

[51] Cristina Nader Vasconcelos and Bodo Rosenhahn. Bipartite graph matching computation on gpu. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 42–55. Springer, 2009. 4

[52] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009. 3

[53] Xuan-Thuy Vo and Kang-Hyun Jo. A review on anchor assignment and sampling heuristics in deep learning-based object detection. *Neurocomputing*, 2022. 3

[54] Xue Yang, Junchi Yan, Qi Ming, Wentao Wang, Xiaopeng Zhang, and Qi Tian. Rethinking rotated object detection with gaussian wasserstein distance loss. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11830–11841. PMLR, 18–24 Jul 2021. 3

[55] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 2, 6, 7

Unbalanced Optimal Transport: A Unified Framework for Object Detection Supplementary Material

A. Optimal Transport Discussion

The *Otpimal Transport* formulation presented throughout the paper is formulated in discrete space. In this section, we present the more general formulation of which the discrete one is a particular case of. We also discuss the Wasserstein distance in the particular context of object detection and the effect of regularization on the uniqueness of the solutions. Only the case of the original OT formulation—or "balanced" case—is covered here.

A.1. Continuous Formulation

More generally, we define Optimal Transport in its continuous form.

Definition 5 (Continuous Optimal Transport)**.** *Given two distributions $\alpha \in \mathscr{P}_+(X)$ and $\beta \in \mathscr{P}_+(Y)$ of same mass $\int \alpha \,\mathrm{d}x = \int \beta \,\mathrm{d}y$, and given an underlying cost function $c : X \times Y \to [0, +\infty]$, we define* Continuous Optimal Transport *as the minimization of a transport cost*

$$\inf \left\{ \int_{X \times Y} c \,\mathrm{d}\gamma : \gamma \in U(\alpha, \beta) \right\}, \tag{8}$$

with admissible solutions, here called transport plans

$$U(\alpha, \beta) = \left\{ \gamma \in \mathscr{P}_+(X \times Y) : \int_Y \mathrm{d}\gamma = \alpha \quad \textit{and} \quad \int_X \mathrm{d}\gamma = \beta \right\}. \tag{9}$$

If a minimum exists, it is called the optimal transport plan $\hat{\gamma}$.

We replace the probability simplex Δ^N by the space on probability distributions $\mathscr{P}_+(X)$ on X. The transport plans are the set of joint probability distribution $\gamma \in \mathscr{P}_+(X \times Y)$, whose marginal distributions are α and β. The discrete formulation (Definition 1) is a particular case where $\alpha = \sum_i \alpha_i \delta_{\hat{\boldsymbol{y}}_i}$, $\beta = \sum_j \beta_j \delta_{\boldsymbol{y}_j}$ and the cost $c = \mathcal{L}_{\text{match}}$. In this case, a minimum always exists.

A.2. Wasserstein Distance

This infimum defines a distance between α and β, called the Wasserstein distance $\mathcal{W}_p(\alpha, \beta)$, provided that the underlying cost function is also a distance $c = d^p$ up to some exponent $p \in [1, +\infty[$. In our case, $\mathcal{L}_{\text{match}}$ is not a distance. More formally, sum of distances are distances. The ℓ^1 norm is a distance, and $1 - \text{IoU}$, or $1 - \text{GIoU}$ also are [7]. However, the cross entropy or the focal loss do not satisfy the triangular inequality or the symmetry properties. In consequence, we cannot talk about a Wasserstein distance here.

Furthermore, interpreting a Wasserstein distance $\mathcal{W}_p(\alpha, \beta)$ would not make much sense even if the underlying matching cost was to be a distance. Indeed, the distributions α and β would be the same at every iteration in our framework. In other words, the distance would always be computed between the same points, but the underlying cost would change and it would be different for each image. Each iteration would be computing the distance of two same points in a changing geometry and each image would have its own evolving geometry.

For completeness, we must mention that the regularized version does not define a distance as $\mathcal{W}_{p,\text{reg.}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}) = -\epsilon \,\mathrm{H}(\boldsymbol{I}_{N_p,N_p}/N_p) > 0$ with $\boldsymbol{I}_{N_p,N_p}$ the identity matrix of size N_p (we refer to [5, 4, 3] for a broader discussion on the subject).

A.3. Uniqueness

We consider here the discrete formulation used throughout the paper. By classical linear programming theory, the non-regularized problem admits a non-unique solution if and only if multiple extreme points minimize the problem. In that case, the set of minimizers is all the linear interpolations between those extreme points. The regularization term however is ϵ-strongly convex; the regularized problem thus always has a unique solution [6].

B. Proofs of the Propositions

In this section, we provide the proofs of Propositions 1 and 2 and enrich them with some insight through a few additional results.

B.1. Hungarian Algorithm

Before providing a proof of the particular equivalence between OT and BM, we first consider a more general result.

Lemma 1. *We consider the rational probability simplex* $\Delta_{\mathbb{Q}}^N = \{\boldsymbol{u} \in \mathbb{Q}_{\geq 0}^N | \sum_i u_i = 1\}$. *Given an OT problem (Definition 1) with underlying distributions* $\boldsymbol{\alpha} \in \Delta_{\mathbb{Q}}^N$ *and* $\boldsymbol{\beta} \in \Delta_{\mathbb{Q}}^M$. *Each extreme point of* $\mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ *is comprised of elements, which are multiples of the* common measure *of* $\boldsymbol{\alpha}$ *and* $\boldsymbol{\beta}$*:*

$$\boldsymbol{P} \text{ is an extreme point of } \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \quad \Longrightarrow \quad \boldsymbol{P} \in \mathrm{CM}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \cdot \mathbb{N}_{\geq 0}^{N \times M}, \tag{10}$$

where the common measure *is the greatest rational such that all non-zero elements of both distributions are multiples of it:*

$$\mathrm{CM}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\mathrm{GCD}\left(\mathrm{LCM}\left([\,\boldsymbol{\alpha},\, \boldsymbol{\beta}\,]\right) / [\,\boldsymbol{\alpha},\, \boldsymbol{\beta}\,]\right)}{\mathrm{LCM}\left([\,\boldsymbol{\alpha},\, \boldsymbol{\beta}\,]\right)} \in \mathbb{Q}_{>0}, \tag{11}$$

with $\mathrm{GCD} : \mathbb{N}_{>0}^N \to \mathbb{N}_{>0}$ *the greatest common divisor and* $\mathrm{LCM} : \mathbb{N}_{>0}^N \to \mathbb{N}_{>0}$ *the lowest common multiple.*

The common measure *extends the* GCD *to non-integers. As an example* $\mathrm{CM}([\,2/3,\, 4/5\,]) = 2/15$ *and* $\mathrm{CM}([\,2/3,\, 5/6,\, 4/7\,]) = 1/42$.

Proof. In [1], Corollary 8.1.3, an algorithm is given to build the exhaustive list of extreme points. It comprises only minimum and subtraction operations, which leave the common measure unchanged. □

Corollary 1. *Given the underlying distributions as in Proposition 1, the extreme points of* $\mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ *are comprised only of zeros and* $1/N_p$*:*

$$\boldsymbol{P} \text{ is an extreme point of } \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \quad \Longrightarrow \quad \boldsymbol{P} \in \{0, 1/N_p\}^{N_p \times (N_g+1)}. \tag{12}$$

This is a direct consequence of Lemma 1 and the mass constraints directly implying that $P_i \leq 1/N_p$ for all i. In this particular case, there is also an equivalence.

Lemma 2. *Given the underlying distributions as in Proposition 1, the extreme points of* $\mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ *are comprised only of zeros and* $1/N_p$*:*

$$\boldsymbol{P} \text{ is an extreme point of } \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \quad \Longleftrightarrow \quad \boldsymbol{P} \in \{0, 1/N_p\}^{N_p \times (N_g+1)} \quad \text{and} \quad \boldsymbol{P} \in \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta}). \tag{13}$$

Proof. We consider Corollary 1 and add the fact that such a match $\boldsymbol{P} \in \{0, 1/N_p\}^{N_p \times (N_g+1)}$ only has one element per row (or prediction if we prefer) to satisfy the mass constraints. Therefore, it cannot be any interpolation of two other extreme points. □

We however also give a more direct proof, based essentially on the same arguments.

Proof. We will first show that the elements of the match $\boldsymbol{P}$ corresponding to any extreme point, can only be $1/N_p$ or 0. Therefore we can consider the associated bipartite graph of the problem: each prediction consists in a node i and each ground truth a node j. Each non-zero value entry of $\boldsymbol{P}$ connects nodes i and j with weight $P_{i,j}$. The solution is admissible if and only if the weight of each node i equals α_i and j equals β_j. A transport plan $\boldsymbol{P}$ is an extreme point if and only if the corresponding bipartite graph only consists in trees, or equivalently, it has no cycle (Theorem 8.1.2 of [1]).

Because the mass constraint must all sum up to one for the predictions, we already know that $P_{i,j} \leq 1/N_p$. We will now proceed *ad absurdum* and suppose that there were to be an entry $0 < P_{i,j} < 1/N_p$ connecting a prediction and a ground truth. In order to satisfy the mass constraints, they would both also have to be connected to another prediction and another ground truth. Similarly, these would also have to be connected to at least one prediction and one ground truth, and so on. They would all form a same graph, or be "linked" together in other words. By consequence, each new connection must be done to yet "unlinked" prediction and ground truth to avoid the formation of a cycle. Considering that there are N_p predictions, there would be at the end at least $2N_p$ edges within the graph. This is incompatible with the fact that there cannot be any cycle (Corollary 8.1.3 of [1]). By consequence, the entries of $\boldsymbol{P}$ must be either 0 or $1/N_p$. □

We can now proceed to prove the said proposition.

Proposition 1. *The Hungarian algorithm with N_p predictions and $N_g \leq N_p$ ground truth objects is a particular case of OT with $\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \subset \mathbb{R}^{N_p \times (N_g+1)}$, consisting of the predictions and the ground truth objects, with the background added $\{\boldsymbol{y}_j\}_{j=1}^{N_g+1} = \{\boldsymbol{y}_j\}_{j=1}^{N_g} \cup \left(\boldsymbol{y}_{N_g+1} = \varnothing\right)$. The chosen underlying distributions are*

$$\boldsymbol{\alpha} = \frac{1}{N_p}[\underbrace{1,\ 1,\ 1,\ \ldots,\ 1}_{N_p \text{ predictions}}], \tag{14}$$

$$\boldsymbol{\beta} = \frac{1}{N_p}[\underbrace{1,\ 1,\ \ldots,\ 1}_{N_g \text{ ground truth objects}},\ \underbrace{(N_p - N_g)}_{\text{background } \varnothing}], \tag{15}$$

provided the background cost is constant: $\mathcal{L}_{match}(\hat{\boldsymbol{y}}_i, \varnothing) = c_\varnothing$. In particular for $j \in [\![N_g]\!]$, we have $\hat{\sigma}(j) = \{i : P_{i,j} \neq 0\}$, or equivalently $\hat{\sigma}(j) = \{i : P_{i,j} = 1/N_p\}$.

Proof. We will demonstrate that OT with $\boldsymbol{\alpha} = \frac{1}{N_p}[\ 1,\ 1,\ 1,\ \ldots,\ 1\ ]$ and $\boldsymbol{\beta} = \frac{1}{N_p}[\ 1,\ 1,\ \ldots,\ 1,\ (N_p - N_g)\ ]$ and constant background cost necessarily has the BM as minimal solution. We first observe that because of the linear nature of the problem, there is at least one extreme point that minimizes the total cost. By directly applying Lemma 2, there must be exactly one match per prediction and exactly one match for each non-background ground truth to satisfy the mass constraints. The added background ground truth has $N_p - N_g$ matches. This is equivalent to saying that disregarding the background ground truth, we wave $\sigma \in \mathcal{P}_{N_g}([\![N_p]\!])$ with $\hat{\sigma}(j) = \{i : P_{i,j} = 1/N_p\}$. The proof is concluded by observing that the part of the background in the total transport cost is equal to $\frac{1}{N_p}(N_p - N_g)\, c_\varnothing$ and is constant, hence not influencing the minimum. □

B.2. Minimum Matching with Threshold

Proposition 2 (Matching to the closest)**.** *We consider the same objects as Proposition 1. In the limit of $\tau_1 \to \infty$ and $\tau_2 = 0$, Unbalanced OT (Definition 4) without regularization ($\epsilon = 0$) admits as solution each prediction being matched to the closest ground truth object unless that distance is greater than a threshold value $\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_{N_g+1} = \varnothing\right) = c_\varnothing$. It is then matched to the background $\varnothing$. In particular, we have*

$$\hat{P}_{i,j} = \begin{cases} \frac{1}{N_p} & \text{if } j = \arg\min_{j \in [\![N_g+1]\!]} \{\mathcal{L}_{match}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j)\}, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Proof. By taking the limit of $\tau_1 \to +\infty$ and setting $\epsilon, \tau_2 = 0$, the problem becomes

$$\begin{aligned} \arg\min \quad & \left\{ \textstyle\sum_{i,j=1}^{N_p, N_g+1} P_{i,j} \mathcal{L}_{\text{match}}(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_j) \,\middle|\, \boldsymbol{P} \in \mathbb{R}_{\geq 0}^{N_p \times (N_g+1)} \right\}, \\ \text{s.t.} \quad & \textstyle\sum_j P_{i,j} = 1/N_p \qquad \forall i. \end{aligned} \tag{17}$$

We can now see that the choice made in each row is independent from the other rows. In other words, each ground truth object can be matched independently of the others. The minimization is then obtained if, for each prediction (or row), all the weight is put on the ground truth object with minimum cost, including the background. This leads to Eq. (16). □

Corollary 2 (Matching to the closest without threshold)**.** *Provided the background cost is more expensive than any other cost $c_\varnothing > \max\left\{\mathcal{L}_{match}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j) \,\middle|\, i \in [\![N_p]\!] \text{ and } j \in [\![N_g]\!]\right\}$, each prediction will always be matched to the closest ground truth.*

In theory, this a much too strong condition, the background cost can just be greater than the minimum cost for each prediction $\mathcal{L}_{\text{match}}(\hat{\boldsymbol{y}}_i, \varnothing) > \min_j \{\mathcal{L}_{\text{match}}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j)\}$. In practice, however, this does not change much. It suffices to set the background cost high enough and we are assured to get a minimum. One could also imagine a different background cost for each prediction in order to have a more granular threshold.

C. Scaling Algorithms

We present here the two scaling algorithms: Sinkhorn's algorithm for "'balanced" *Optimal Transport* and its variant for *Unbalanced Optimal Transport*. We further show how it is connected to the softmax.

C.1. Sinkhorn and Variant

These two algorithms are taken from [6, 2]. In particular we can see how taking $\tau_1 \to +\infty$ and $\tau_2 \to +\infty$ in Algorithm 2 leads to Algorithm 1. Indeed, we have $\lim_{\tau \to +\infty} \frac{\tau}{\tau+\epsilon} = 1$. By $\oslash$, we denote the element-wise (or Hadamard) division.

Data: Distributions $\boldsymbol{\alpha} \in \Delta^{N_p}$ and $\boldsymbol{\beta} \in \Delta^{N_g+1}$, regularization parameter $\epsilon \in \mathbb{R}_{>0}$ and cost matrix $\boldsymbol{C} = \left[\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right]_{i,j=1}^{N_p,N_g+1} \in \mathbb{R}_{\geq 0}^{N_p \times N_g+1}$ (including background $\boldsymbol{y}_{N_g+1} = \varnothing$).
Result: Match $\hat{\boldsymbol{P}} \in \Delta^{N_p,N_g+1}$.

begin
- $\boldsymbol{K}_\epsilon \longleftarrow \exp\left(-\boldsymbol{C}/\epsilon\right)$ /* Gramm matrix (element-wise) */
- $\boldsymbol{u} \longleftarrow \mathbf{1}_{N_p}/N_p$ /* Dual variable associated with $\boldsymbol{\alpha}$ */
- $\boldsymbol{v} \longleftarrow \mathbf{1}_{N_g+1}/\left(N_g+1\right)$ /* Dual variable associated with $\boldsymbol{\beta}$ */
- **repeat**
  - $\boldsymbol{u} \longleftarrow \boldsymbol{\alpha} \oslash \left(\boldsymbol{K}_\epsilon \boldsymbol{v}\right)$ /* Scaling iteration for $\boldsymbol{u}$ */
  - $\boldsymbol{v} \longleftarrow \boldsymbol{\beta} \oslash \left(\boldsymbol{K}_\epsilon^\top \boldsymbol{u}\right)$ /* Scaling iteration for $\boldsymbol{v}$ */
- **until** *convergence*
- $\hat{\boldsymbol{P}} \longleftarrow \boldsymbol{u}\boldsymbol{K}_\epsilon\boldsymbol{v}$

Algorithm 1: Sinkhorn's algorithm for "balanced" *Optimal Transport* with regularization.

Data: Distributions $\boldsymbol{\alpha} \in \Delta^{N_p}$ and $\boldsymbol{\beta} \in \Delta^{N_g+1}$, regularization parameter $\epsilon \in \mathbb{R}_{>0}$, constraint parameters $\tau_1, \tau_2 \in \mathbb{R}_{\geq 0}$ and cost matrix $\boldsymbol{C} = \left[\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right]_{i,j=1}^{N_p,N_g+1} \in \mathbb{R}_{\geq 0}^{N_p \times N_g+1}$ (including background $\boldsymbol{y}_{N_g+1} = \varnothing$).
Result: Match $\hat{\boldsymbol{P}} \in \mathbb{R}_{\geq 0}^{N_p,N_g+1}$.

begin
- $\boldsymbol{K}_\epsilon \longleftarrow \exp\left(-\boldsymbol{C}/\epsilon\right)$ /* Gramm matrix (element-wise) */
- $\boldsymbol{u} \longleftarrow \mathbf{1}_{N_p}/N_p$ /* Dual variable associated with $\boldsymbol{\alpha}$ */
- $\boldsymbol{v} \longleftarrow \mathbf{1}_{N_g+1}/\left(N_g+1\right)$ /* Dual variable associated with $\boldsymbol{\beta}$ */
- **repeat**
  - $\boldsymbol{u} \longleftarrow \left(\boldsymbol{\alpha} \oslash \left(\boldsymbol{K}_\epsilon \boldsymbol{v}\right)\right)^{\frac{\tau_1}{\tau_1+\epsilon}}$ /* Scaling iteration for $\boldsymbol{u}$ */
  - $\boldsymbol{v} \longleftarrow \left(\boldsymbol{\beta} \oslash \left(\boldsymbol{K}_\epsilon^\top \boldsymbol{u}\right)\right)^{\frac{\tau_2}{\tau_2+\epsilon}}$ /* Scaling iteration for $\boldsymbol{v}$ */
- **until** *convergence*
- $\hat{\boldsymbol{P}} \longleftarrow \boldsymbol{u}\boldsymbol{K}_\epsilon\boldsymbol{v}$

Algorithm 2: Scaling algorithm for *Unbalanced Optimal Transport* with regularization.

C.2. Connection with the Softmax

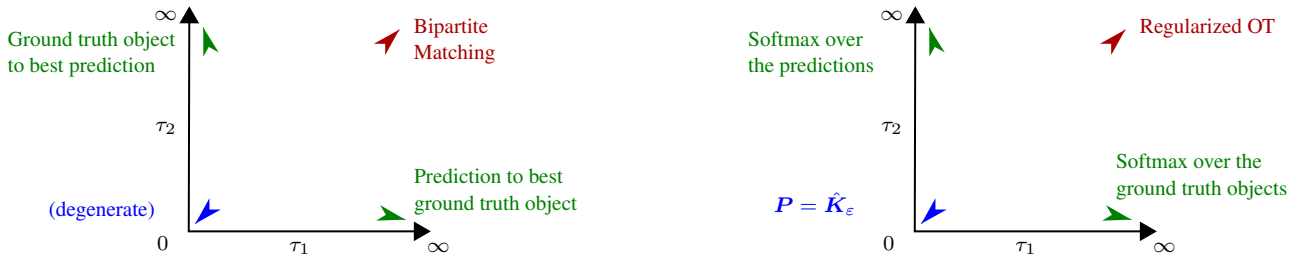
In this section, we lay a connection between the softmax and the solutions of the scaling algorithms, in particular considering its first iterations. We consider more precisely the softmin, which is the opposite of the softmax: $\left(\text{softmin}(\boldsymbol{v})\right)_i = \left(\text{softmax}(-\boldsymbol{v})\right)_i = \exp(-v_i)/\sum_{j=1}^N \exp(-v_j)$, for any vector $\boldsymbol{v} \in \mathbb{R}^N$. Considering a softmin over $\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)$ is thus the same as considering the softmax over $-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)$, as in [8]. By simplicity, we will only use the softmax terminology.

C.2.1 Without Background

We first consider the case without background, where the underlying distributions are equal to $\boldsymbol{\alpha} = \mathbf{1}_{N_p}/N_p$ and $\boldsymbol{\beta} = \mathbf{1}_{N_g}/N_g$. This does not correspond to the setup of Prop. 1 and only approximates a one-to-one match if $N_p = N_g$.

Proposition 3. *Consider the two uniform distributions $\boldsymbol{\alpha} = \mathbf{1}_{N_p}/N_p$ and $\boldsymbol{\beta} = \mathbf{1}_{N_g}/N_g$ with cost $\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)$. The solution of the Unbalanced OT scaling algorithm with regularization $\varepsilon = 1$, $\tau_1 = 0$ and $\tau_2 \to +\infty$ is proportional to performing a softmax over the predictions, for each ground truth object. In particular, we have*

$$\hat{P}_{i,j} = \frac{\exp\left(-\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}{N_g \sum_{i=1}^{N_p} \exp\left(-\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}. \tag{18}$$



(a) No regularization $\varepsilon = 0$ (same as Fig. 4).

(b) With regularization $\varepsilon = 1$.

Figure 7. Comparison of the different limit cases of *Unbalanced Optimal Transport*, with and without regularization.

Proof. We consider the first scaling iteration from Alg. 2. We first observe that the exponents lead to $\lim_{\tau_1 \to 0} \frac{\tau_1}{\tau_1 + \varepsilon} = 0$ and $\lim_{\tau_2 \to +\infty} \frac{\tau_2}{\tau_2 + \varepsilon} = 1$. Starting with $\boldsymbol{v}_0 = \mathbf{1}_{N_g}/N_g$, we obtain the new

$$\boldsymbol{u}_1 = \left(\boldsymbol{\alpha} \oslash \left(\boldsymbol{K}_\varepsilon \boldsymbol{v}_0\right)\right)^0 = \mathbf{1}_N, \qquad \text{or } \left(\boldsymbol{u}_1\right)_i = 1, \tag{19}$$

$$\boldsymbol{v}_1 = \left(\boldsymbol{\beta} \oslash \left(\boldsymbol{K}_\varepsilon^\top \boldsymbol{u}_1\right)\right)^1 = \left(\mathbf{1}_{N_g}/N_g\right) \oslash \left(\boldsymbol{K}_\varepsilon^\top \mathbf{1}_N\right), \qquad \text{or } \left(\boldsymbol{v}_1\right)_j = \frac{1}{N_g \sum_{i=1}^{N_p} \exp\left(-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}. \tag{20}$$

We observe that $\boldsymbol{u}_2 = \boldsymbol{u}_1$ and $\boldsymbol{v}_2 = \boldsymbol{v}_1$ and conclude that the algorithm converges after only one iteration. Computing the match $\hat{\boldsymbol{P}} = \boldsymbol{u}\boldsymbol{K}_\epsilon\boldsymbol{v}$ leads to the softmax. □

The exact opposite happens if we consider $\tau_1 \to +\infty$ and $\tau_2 = 0$ instead: the softmax is taken over the ground truth objects for each prediction. The proof is the same, just inverting $\boldsymbol{u}$ and $\boldsymbol{v}$ and obtaining factor $1/N_p$ instead. This can be observed at Fig. 7b.

If we would like to exactly obtain the softmax without the factor $1/N_g$ (or $1/N_p$), we could consider only one iteration starting with both initial dual variables $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$. It would however not be the optimal match $\hat{\boldsymbol{P}}$ and will converge to the same solution as in Prop. 3 after the second—and last—iteration. Nevertheless, starting from both initial dual variables is more interesting in the "balanced" case.

Proposition 4. *Consider the two uniform distributions $\boldsymbol{\alpha} = \mathbf{1}_{N_p}/N_p$ and $\boldsymbol{\beta} = \mathbf{1}_{N_g}/N_g$ with cost $\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)$. Starting from both initial dual variables, one iteration of the "balanced" OT scaling algorithm with regularization $\varepsilon = 1$ is equal to*

$$P_{i,j} = \frac{\exp\left(-\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}{\sum_{i=1}^{N_p} \exp\left(-\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right) \cdot \sum_{j=1}^{N_g} \exp\left(-\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}. \tag{21}$$

Proof. We consider the first scaling iteration from Alg. 1 with $\boldsymbol{\alpha} = \mathbf{1}_{N_p}/N_p$ and $\boldsymbol{\beta} = \mathbf{1}_{N_g}/N_g$. Starting with $\boldsymbol{u}_0 = \mathbf{1}_{N_p}/N_p$ and $\boldsymbol{v}_0 = \mathbf{1}_{N_g}/N_g$, we obtain the new

$$\boldsymbol{u}_1 = \boldsymbol{\alpha} \oslash \left(\boldsymbol{K}_\varepsilon \boldsymbol{v}_0\right) = \left(\mathbf{1}_{N_p}/N_p\right) \oslash \left(\boldsymbol{K}_\varepsilon \left(\mathbf{1}_{N_g}/N_g\right)\right), \qquad \text{or } \left(\boldsymbol{u}_1\right)_i = \frac{N_g}{N_p \sum_{j=1}^{N_g} \exp\left(-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}, \tag{22}$$

$$\boldsymbol{v}_1 = \boldsymbol{\beta} \oslash \left(\boldsymbol{K}_\varepsilon^\top \boldsymbol{u}_0\right) = \left(\mathbf{1}_{N_g}/N_g\right) \oslash \left(\boldsymbol{K}_\varepsilon^\top \left(\mathbf{1}_{N_p}/N_p\right)\right), \qquad \text{or } \left(\boldsymbol{v}_1\right)_j = \frac{N_p}{N_g \sum_{i=1}^{N_p} \exp\left(-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}. \tag{23}$$

Computing the match $\boldsymbol{P} = \boldsymbol{u}\boldsymbol{K}_\epsilon\boldsymbol{v}$ leads to the Eq. 21. This is not the optimal match $\hat{\boldsymbol{P}}$ as the algorithm did not converge yet. □

The *dual-softmax* considered in [8] is essentially the same as Prop. 4, with the difference of a factor 2 in the numerator's exponential:

$$P_{i,j} = \text{softmax}\left(\left[-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_k\right)\right]_{k=1}^{N_g}\right)_j \cdot \text{softmax}\left(\left[-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_l, \boldsymbol{y}_j\right)\right]_{j=1}^{N_p}\right)_i, \tag{24}$$

$$= \frac{\exp\left(-2\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}{\sum_{i=1}^{N_p} \exp\left(-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right) \cdot \sum_{j=1}^{N_g} \exp\left(-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}. \tag{25}$$

C.2.2 With Background

We now consider the underlying distributions as defined in Prop. 1. Fundamentally, adding a background with a different weight than the other ground truth objects does not change much. The unbalanced case with $\tau_1 \to +\infty$ and $\tau_2 = 0$ remains exactly the same. The opposite case with $\tau_1 = 0$ and $\tau_2 \to +\infty$ now becomes

$$\hat{P}_{i,j} = \frac{1}{N_p} \frac{\exp\left(-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}{\sum_{i=1}^{N_p} \exp\left(-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}, \tag{26}$$

for all $1 \leq j \leq N_g$, and

$$\hat{P}_{i,j} = \frac{N_p - N_g}{N_p} \frac{\exp\left(-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}{\sum_{i=1}^{N_p} \exp\left(-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}, \tag{27}$$

for $j = N_g + 1$ (the background). In essence, this ensures that the mass constraints induced by τ_2 are satisfied, as the background has a higher weight.

Similarly, the "balanced" case is the same as Eq. 21 for all $1 \leq j \leq N_g$. For $j = N_g + 1$, we have the same with an added factor:

$$P_{i,j} = (N_p - N_g) \frac{\exp\left(-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}{\sum_{i=1}^{N_p} \exp\left(-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right) \cdot \sum_{j=1}^{N_g} \exp\left(-\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right)}. \tag{28}$$

C.2.3 Other Regularization

We can also consider other cases that having the regularization $\varepsilon = 1$. The regularization ε controls the "softness" of the softmax: the greater is ε, the softer is the minimum; the smaller, the harder. In the case of no regularization at all ($\varepsilon \to 0$), the softmax is exactly a minimum as proven in Prop. 2. This can be observed at Fig. 8.

D. Scaling the Entropic Parameter

In this section, we consider the particular choice of the entropic regularization parameter. In particular, we study how it scales with the problem size.

D.1. Uniform Matches

Definition 6 (Matches)**.** *We define a* match $\boldsymbol{P} \in \mathbb{R}_+^{N_p \times (N_g+1)}$ *as a positive matrix of unity mass* $\sum_{i,j} P_{i,j} = 1$*. The set of all matches of size* $N_p \times (N_g + 1)$ *is the joint probability simplex* $\Delta^{N_p \times (N_g+1)}$.

We now consider a particular subset of all these matches.

Definition 7 (Uniform Matches)**.** *We define the set of* uniform matches $\Delta_{\text{unif.}}^{N_p \times (N_g+1)} \subsetneq \Delta^{N_p \times (N_g+1)}$ *as the set of matrices* $\boldsymbol{P}^{\text{unif.}} \in \Delta_{\text{unif.}}^{N_p \times (N_g+1)}$*, containing only zero elements and all non-zero elements having the same value:*

$$P_{i,j}^{\text{unif.}} = \begin{cases} 0 & \text{for some values,} \\ 1/\left|\text{spt}\left(\boldsymbol{P}^{\text{unif.}}\right)\right| & \text{for the other values,} \end{cases} \tag{29}$$

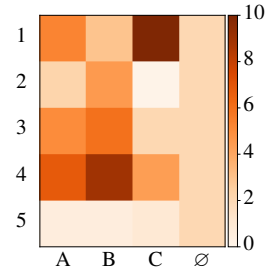
with the support $\text{spt} : \boldsymbol{P} \mapsto \{(i,j) : P_{i,j} \neq 0\}$ *and* $|\cdot|$ *the cardinality of a set.*

We directly see from the definition that the matrices are well defined as they have unity mass. They are uniquely defined by the carnality of their support.

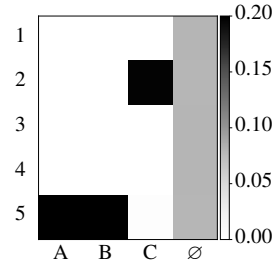
Proposition 5 (Cardinality)**.** *The cardinality of* $\Delta_{\text{unif.}}^{N_p \times (N_g+1)}$ *is given by*

$$\left|\Delta_{\text{unif.}}^{N_p \times (N_g+1)}\right| = 2^{N_p(N_g+1)}. \tag{30}$$

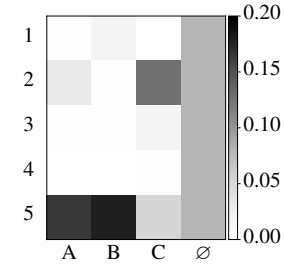
Proof. We first notice that the different possible supports $k = \text{spt}\left(\boldsymbol{P}^{\text{unif.}}\right)$ range from $1 \leq k \leq N_p\left(N_g + 1\right)$. For any support of size k, we have to consider a uniform match containing all combinations. The rest follows from the binomial identity $\sum_{k=1}^{N_p(N_g+1)} \binom{N_p(N_g+1)}{k} = 2^{N_p(N_g+1)}$. □



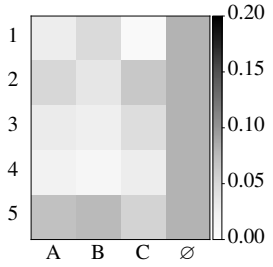
(a) Cost between the predictions and the ground truth objects.



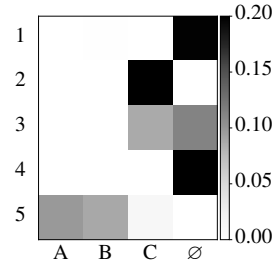
(b) Unbalanced OT with $\varepsilon = 0.2$, $\tau_1 = 0.001$ and $\tau_2 = 1000$.



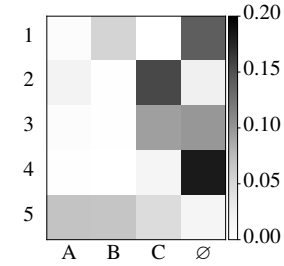
(c) Unbalanced OT with $\varepsilon = 1$, $\tau_1 = 0.001$ and $\tau_2 = 1000$.



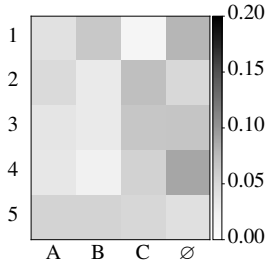
(d) Unbalanced OT with $\varepsilon = 5$, $\tau_1 = 0.001$ and $\tau_2 = 1000$.



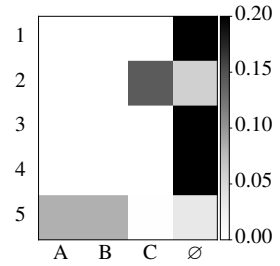
(e) Unbalanced OT with $\varepsilon = 0.2$, $\tau_1 = 1000$ and $\tau_2 = 0.001$.



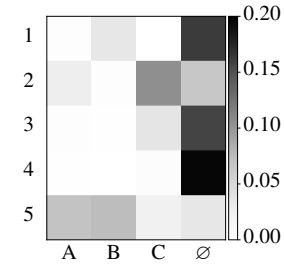
(f) Unbalanced OT with $\varepsilon = 1$, $\tau_1 = 1000$ and $\tau_2 = 0.001$.



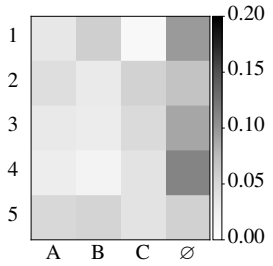
(g) Unbalanced OT with $\varepsilon = 5$, $\tau_1 = 1000$ and $\tau_2 = 0.001$.



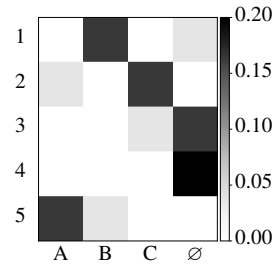
(h) "Balanced" OT with only one iteration and $\varepsilon = 0.2$.



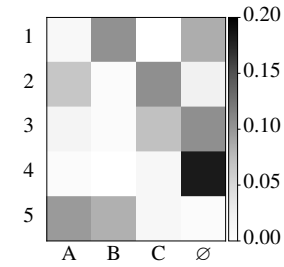
(i) "Balanced" OT with only one iteration and $\varepsilon = 1$.



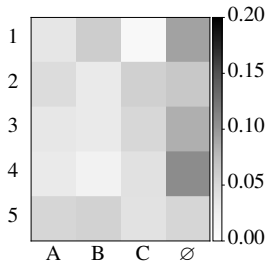
(j) "Balanced" OT with only one iteration and $\varepsilon = 5$.



(k) "Balanced" OT until convergence with $\varepsilon = 0.2$.



(l) "Balanced" OT until convergence with $\varepsilon = 1$.



(m) "Balanced" OT until convergence with $\varepsilon = 5$.

Figure 8. Connection between scaling algorithms and the softmax. The pairwise matching cost between the predictions (numbers) and the ground truth objects (letters) is given in Fig. 8a. The background cost is $c_\varnothing = 2$. The scaling algorithm for Unbalanced OT corresponds to performing the softmax column-wise (Figs. 8b, 8c and 8d), or row-wise (Figs. 8e, 8f and 8g). Similarly, one iteration of the scaling algorithm for "balanced" OT is almost equivalent to the dual-softmax (Figs. 8h, 8i and 8j), but does not satisfy the mass constraints unlike when it is run until convergence (Figs. 8k, 8l and 8m).

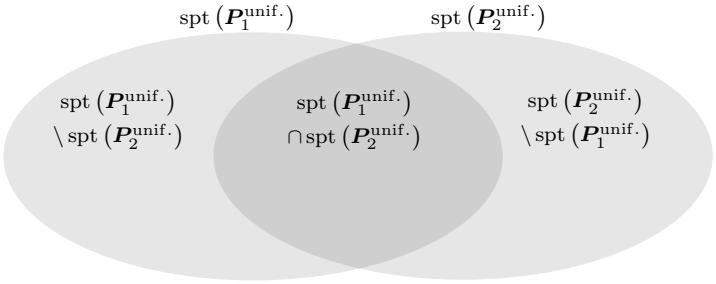


Figure 9. Decomposition of the non-zero indices of two uniform transport matrices $\boldsymbol{P}_1^{\text{unif.}}, \boldsymbol{P}_2^{\text{unif.}} \in \Delta_{\text{unif.}}^{N_p \times \left(N_g+1\right)}$.

We can also see that the uniform matches cover the set of all matches.

Proposition 6 (Diameter)**.** *The diameter of the set of transport matrices $\Delta^{N_p \times (N_g+1)}$ and uniform transport matrices $\Delta_{\text{unif.}}^{N_p \times (N_g+1)}$, equipped with the Fröbenius norm $\|\cdot\|_F$, is given by*

$$\operatorname{diam}\left(\Delta^{N_p \times (N_g+1)}\right) = \operatorname{diam}\left(\Delta_{\text{unif.}}^{N_p \times (N_g+1)}\right) = \sqrt{2} \tag{31}$$

Proof. Maximizing the Fröbenius norm is equivalent to considering the maximization of $\sum_i (u_i - v_i)^2$ subject to $\sum_i u_i = 1$ and $\sum_i v_i = 1$, with $\boldsymbol{u}, \boldsymbol{v} \geq 0$. It takes its maximum value on the boundary of the admissible solutions, for $u_i = 1$ (the rest is zero) and $v_j = 1$ (the rest zero) for any $j \neq i$. These extreme points are also in $\Delta_{\text{unif.}}^{N_p \times (N_g+1)}$, in particular those of unity support $\operatorname{spt}\left(\boldsymbol{P}^{\text{unif.}}\right) = 1$. □

Proposition 7. *The Fröbenius norm square $\|\boldsymbol{P}_1^{\text{unif.}} - \boldsymbol{P}_2^{\text{unif.}}\|_F^2$ between two uniform matches $\boldsymbol{P}_1^{\text{unif.}}, \boldsymbol{P}_2^{\text{unif.}} \in \Delta_{\text{unif.}}^{N_p \times (N_g+1)}$ is given by*

$$\frac{\left|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right)\right| + \left|\operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right| - 2\left|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right) \cap \operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right|}{\left|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right)\right| \left|\operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right|}. \tag{32}$$

Proof. By decomposing the all indices as in Figure 9 in

$$\begin{aligned} \operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right) \cup \operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right) \quad = \quad & \left(\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right) \setminus \operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right) \\ & \cup \left(\operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right) \setminus \operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right)\right) \\ & \cup \left(\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right) \cap \operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right), \end{aligned}$$

and noticing that all other values are zero, we have for $\|\boldsymbol{P}_1^{\text{unif.}} - \boldsymbol{P}_2^{\text{unif.}}\|_F^2$

$$\begin{aligned} & \left(\left|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right)\right| - \left|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right) \cap \operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right|\right) \frac{1}{\left|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right)\right|^2} \\ + \quad & \left(\left|\operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right| - \left|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right) \cap \operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right|\right) \frac{1}{\left|\operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right|^2} \\ + \quad & \left|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right) \cap \operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right| \left(\frac{1}{\left|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right)\right|} - \frac{1}{\left|\operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right|}\right)^2. \end{aligned}$$

The rest is just a simplification of the latter. □

Corollary 3. *Each uniform match $\boldsymbol{P}_1^{\text{unif.}} \in \Delta_{\text{unif.}}^{N_p \times (N_g+1)}$ has as closest neighbors all other uniform matches $\boldsymbol{P}_2^{\text{unif.}} \in \Delta_{\text{unif.}}^{N_p \times (N_g+1)}$ of support increased by one $\left|\operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right| = \left|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right)\right| + 1$ and differing in support for only one entry $\left|\operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right) \setminus \operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right)\right| = 1$. In particular, the square Fröbenius norm is then equal to*

$$\left\|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right) - \operatorname{spt}\left(\boldsymbol{P}_2^{\text{unif.}}\right)\right\|_F^2 = \frac{1}{\left|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right)\right| \left(\left|\operatorname{spt}\left(\boldsymbol{P}_1^{\text{unif.}}\right)\right| + 1\right)}. \tag{33}$$

In in the particular limit case of $\left|\mathrm{spt}\left(\boldsymbol{P}_1^{\mathrm{unif.}}\right)\right| = N_p\left(N_g+1\right)$*, its closest neighbors are all the* $\boldsymbol{P}_2^{\mathrm{unif.}}$ *such that* $\left|\mathrm{spt}\left(\boldsymbol{P}_2^{\mathrm{unif.}}\right)\right| = N_p\left(N_g+1\right)-1$.

Proposition 8. *We consider the projector* $\mathbb{P}:\Delta^{N_p\times(N_g+1)}\to\Delta_{\mathrm{unif.}}^{N_p\times(N_g+1)}$*, that minimizes the Fröbenius norm. For any* $\boldsymbol{P}\in\Delta^{N_p\times(N_g+1)}$*, we consider*

$$\mathbb{P}(\boldsymbol{P}) = \underset{\boldsymbol{P}^{\mathrm{unif.}}\in\Delta_{\mathrm{unif.}}^{N_p\times(N_g+1)}}{\mathrm{argmin}} \|\boldsymbol{P}-\boldsymbol{P}^{\mathrm{unif.}}\|_F. \tag{34}$$

It is given by the matrix $\boldsymbol{P}^{\mathrm{unif.}}\in\Delta_{\mathrm{unif.}}^{N_p\times(N_g+1)}$ *with the* k *greatest elements of* $\boldsymbol{P}$ *as support and*

$$k = \underset{k\in[\![N_p(N_g+1)]\!]}{\arg\max}\ \frac{1}{k}\left(2\sum_{\substack{k\ \text{greatest}\\ \text{elements}}} P_{ij} - 1\right). \tag{35}$$

Proof. We consider the distance between any element $\boldsymbol{P}\in\Delta^{N_p\times(N_g+1)}$ and $\boldsymbol{P}^{\mathrm{unif.}}\in\Delta_{\mathrm{unif.}}^{N_p\times(N_g+1)}$: $\|\boldsymbol{P}-\boldsymbol{P}^{\mathrm{unif.}}\|_F^2 = \sum_{i,j=1}^{N_p,(N_g+1)}\left(P_{i,j}-P_{i,j}^{\mathrm{unif.}}\right)^2 = \sum_{i,j=1}^{N_p,(N_g+1)} P_{i,j}^2 + \left(P_{i,j}^{\mathrm{unif.}}\right)^2 - 2P_{ij}P_{i,j}^{\mathrm{unif.}}$. We notice that because of the uniform nature of $\boldsymbol{P}^{\mathrm{unif.}}$, it only has k non-zero elements, all equal to $1/k$. By consequence we have $\sum_{i,j=1}^{N_p,(N_g+1)}\left(P_{ij}^{\mathrm{unif.}}\right)^2 = \frac{1}{k}$ and $\sum_{i,j=1}^{N_p,(N_g+1)} P_{ij}P_{ij}^{\mathrm{unif.}} = \frac{1}{k}\sum_{\mathrm{spt}(\boldsymbol{P}^{\mathrm{unif.}})} P_{ij}$. The distance is now equal to $\|\boldsymbol{P}-\boldsymbol{P}^{\mathrm{unif.}}\|_F^2 = \|\boldsymbol{P}\|_F^2 + \frac{1}{k} - \frac{2}{k}\sum_{\mathrm{spt}(\boldsymbol{P}')} P_{ij}$, and is minimal if $\frac{2}{k}\sum_{\mathrm{spt}(\boldsymbol{P}^{\mathrm{unif.}})} P_{ij} - \frac{1}{k}$ is maximal which is unique as it suffices to see that is it reached once

$$\sum_{\substack{k\ \text{greatest}\\ \text{elements}}} P_{ij} > P_{\substack{\text{next}\\ \text{greatest}}} + \frac{1}{2}, \tag{36}$$

is satisfied. □

The norm with the projector is therefore also given by $\|\boldsymbol{P}-\mathbb{P}(\boldsymbol{P})\|_F^2 = \|\boldsymbol{P}\|_F - 2\sum_{\mathrm{spt}(\mathbb{P}(\boldsymbol{P}))} P_{ij} + \frac{1}{|\mathrm{spt}(\mathbb{P}(\boldsymbol{P}))|}$. Equation (36) gives a direct algorithm to determine k and thus the projected value of any match $\boldsymbol{P}$.

D.2. Entropy

The study of uniform matches is relevant as they have an easy formulation for their entropy.

Definition 8 (Entropy)**.** *The entropy* $\mathrm{H}:\Delta^{N_p\times(N_g+1)}\to\mathbb{R}_{\geq 0}$ *of a match* $\boldsymbol{P}$ *is given by*

$$\mathrm{H}(\boldsymbol{P}) \overset{def.}{=} -\sum_{i,j} P_{i,j}\left(\log\left(P_{i,j}\right)-1\right). \tag{37}$$

If one of the elements would be zero, i.e., $P_{i,j}=0$*, we consider* $P_{i,j}\log\left(P_{i,j}-1\right)=0$.

The latter condition ensures that the entropy is well defined. This choice is justified as it remains consistent with the limit. Some authors prefer another convention [6].

Lemma 3. *The entropy of a uniform match* $\boldsymbol{P}^{\mathrm{unif.}}\in\Delta_{\mathrm{unif.}}^{N_p\times(N_g+1)}$ *is given by*

$$\mathrm{H}(\boldsymbol{P}^{\mathrm{unif.}}) = \log\left(\left|\mathrm{spt}(\boldsymbol{P}^{\mathrm{unif.}})\right|\right)+1. \tag{38}$$

Proof. The proof is a direct application of the definition of the entropy:

$$\begin{aligned}
\mathrm{H}(\boldsymbol{P}^{\mathrm{unif.}}) &= -\sum_{i,j} P_{i,j}^{\mathrm{unif.}}\left(\log\left(P_{i,j}^{\mathrm{unif.}}\right)-1\right), && (39)\\
&= -\sum_{\mathrm{spt}(\boldsymbol{P}^{\mathrm{unif.}})} P_{i,j}^{\mathrm{unif.}}\left(\log\left(P_{i,j}^{\mathrm{unif.}}\right)-1\right), && (40)\\
&= -\frac{\left|\mathrm{spt}\left(\boldsymbol{P}^{\mathrm{unif.}}\right)\right|}{\left|\mathrm{spt}\left(\boldsymbol{P}^{\mathrm{unif.}}\right)\right|}\left(\log\left(\frac{1}{\left|\mathrm{spt}\left(\boldsymbol{P}^{\mathrm{unif.}}\right)\right|}\right)-1\right), && (41)\\
&= \log\left(\left|\mathrm{spt}(\boldsymbol{P}^{\mathrm{unif.}})\right|\right)+1. && (42)
\end{aligned}$$

□

Proposition 9. *For any match* $\boldsymbol{P} \in \Delta^{N_p \times (N_g+1)}$,

$$1 \leq H(\boldsymbol{P}) \leq \log(N_p\,(N_g+1)) + 1. \tag{43}$$

Proof. For an arbitrary coupling matrix, the entropy is always minimal if $P_{i,j} = 1$ for one element and all the others are zero. Similarly, the entropy is always for the uniform match $P_{i,j} = 1/\left|\mathrm{spt}\left(\boldsymbol{P}\right)\right|$ for all i, j, with $\left|\mathrm{spt}\left(\boldsymbol{P}\right)\right| = N_p \times (N_g+1)$. □

D.3. Rule of Thumb

We first consider two different matches of different dimensions $\boldsymbol{P}_1 \in \Delta^{N_{p,1} \times (N_{g,1}+1)}$ and $\boldsymbol{P}_2 \in \Delta^{N_{p,2} \times (N_{g,2}+1)}$. In this case, the OT with regularization cost (Definition 3) is given by $\sum_{i,j=1}^{N_p,(N_g+1)} P_{i,j}\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right) - \epsilon\,\mathrm{H}(\boldsymbol{P})$. The goal is to scale the regularization parameter ϵ in such a way that the weight of the entropy is proportionally the same. Because of unit mass of any match, we could assume that the first term $\sum_{i,j=1}^{N_p,(N_g+1)} P_{i,j}\mathcal{L}_{\text{match}}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)$ is independent of N_p and N_g in magnitude. We therefore have to guarantee that $\epsilon_1 \mathrm{H}\left(\boldsymbol{P}_1\right) = \epsilon_2 \mathrm{H}\left(\boldsymbol{P}_2\right)$. Given an already determined regularization value ϵ_1 for one of the two sizes, the other can be found with $\epsilon_2 = \epsilon_1 \mathrm{H}\left(\boldsymbol{P}_1\right) / \mathrm{H}\left(\boldsymbol{P}_2\right)$. In practice, however, the entropy is not trivial and we can rely on the projection onto the uniform matches

$$\epsilon_1 = \epsilon_2 \frac{\log\left(\left|\mathrm{spt}\left(\mathbb{P}\left(\boldsymbol{P}_2\right)\right)\right|\right) + 1}{\log\left(\left|\mathrm{spt}\left(\mathbb{P}\left(\boldsymbol{P}_1\right)\right)\right|\right) + 1}. \tag{44}$$

In the particular case of Proposition 1, we can use the approximation $\left|\mathrm{spt}\left(\mathbb{P}\left(\boldsymbol{P}\right)\right)\right| = N_p$, which gives

$$\epsilon_1 = \epsilon_2 \frac{\log\left(N_{p,2}\right) + 1}{\log\left(N_{p,1}\right) + 1}. \tag{45}$$

The idea is to determine the optimal value ϵ_1 on toy examples. By setting $N_p = N_{p,2}$, $\epsilon = \epsilon_2$ and $\epsilon_0 = \epsilon_1\left(\log\left(N_{p,1}\right) + 1\right)$, we can use the simple scaling formula $\epsilon = \epsilon_0 / \left(\log\left(N_p\right) + 1\right)$. From our experiments, we determined $\epsilon_0 = 0.12$.

E. Qualitative Analysis

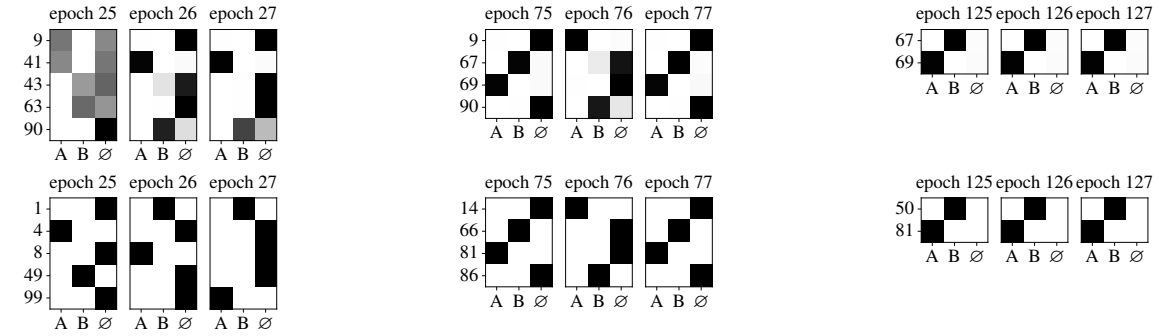
This section provides qualitative examples (Figure 10 and Figure 11) of some matches, as well as a convergence analysis for DETR and Deformable DETR. We compare the losses and matches $\boldsymbol{P}$ of the two matching algorithms at different training epochs.

Figure 10 shows some assignments of the two matching algorithms for DETR on the Color Boxes dataset. We sample examples with few ground truth objects for readability. We only show predictions that are matched at least once with a background $\varnothing$ ground truth in three consecutive epochs. At the beginning of the training, the *Bipartite Matching* with the *Hungarian algorithm* assigns different predictions to the ground truth objects from one epoch to the other. As an example, the algorithm for image №630 assigns predictions $\{4, 49\}$, $\{8, 1\}$ and then $\{99, 1\}$ to the ground-truth objects $\{A, B\}$ at epoch 25 to 27 (Figure 10a). The regularized OT match instead provides a smoother solution and is more consistent from one epoch to the other. Later in training, Figure 10a illustrates that the regularized OT matches are one-to-one and behave like the bipartite ones.

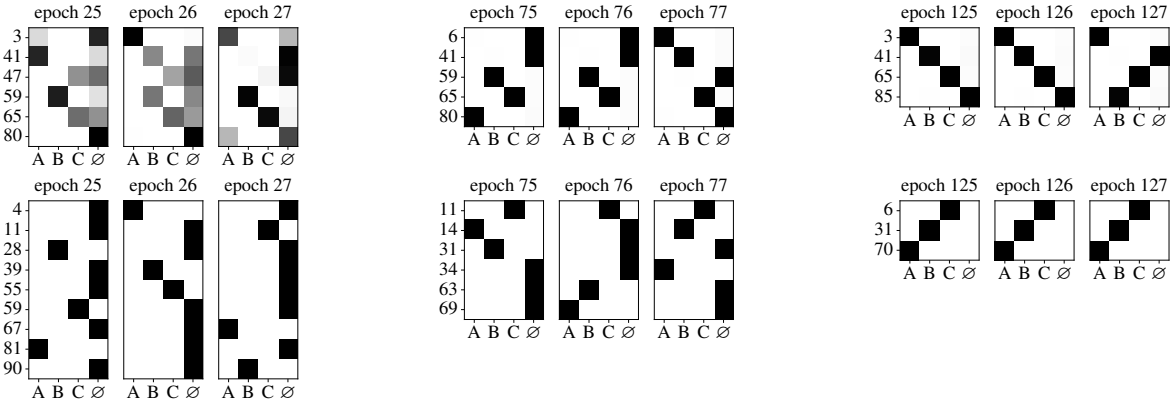
Figure 12 provides the loss curves for DETR on the Color Boxes dataset. The curves suggest that the cross-entropy loss term mainly drives the convergence speedup in the early training epochs. We don't observe such speedups on COCO or with Deformable DETR (Figure 13). An explanation could be that the difference between DETR and Deformable DETR is due to the slower convergence of transformers (we also tried DETR with the focal loss from Deformable DETR without improvement). The difference between Color Boxes and COCO is difficult to isolate, but probably due to the wider class diversity in the latter.

F. Number of Sinkhorn Iterations

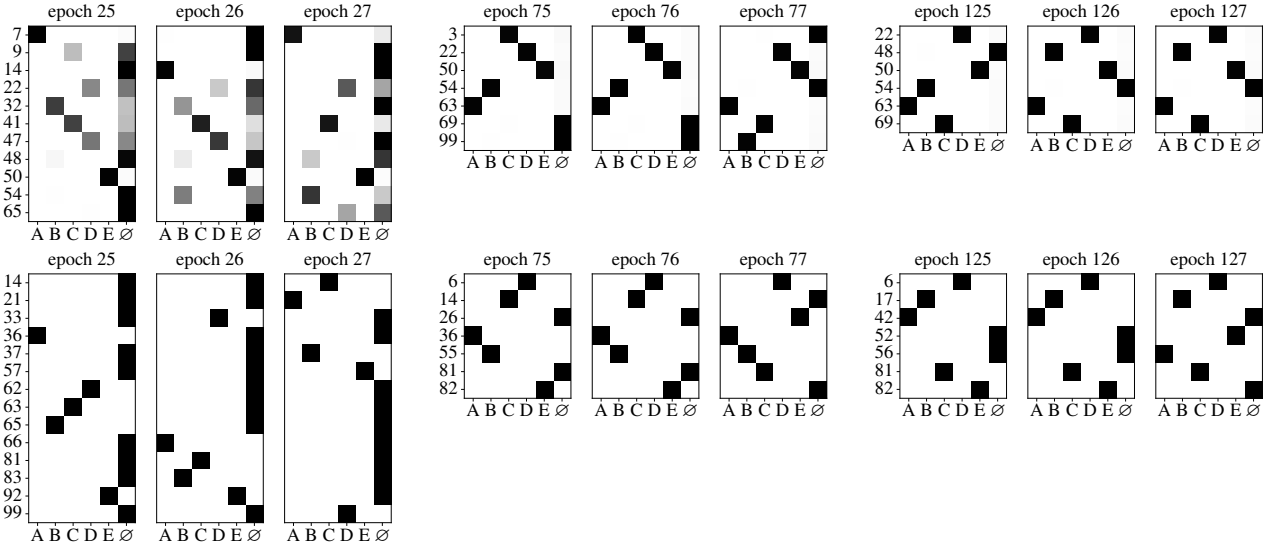
Using a stopping criterion is not straightforward when solving a batch of matching problems. The scaling algorithm is therefore set to a fixed number of iterations. Figure 14 displays the results for different numbers of iterations. For the balanced OT with 300 predictions (Figure 14a), the AP increases only slightly when more than 10 iterations are performed. Furthermore, it is sufficient to run 1 iteration in terms of the AR. For the *Unbalanced OT* with 8,732 predictions (Figure 14b), the metrics are significantly lower when running for less than 5 iterations. Again, running more than 10 iterations only slightly improves the final performance. This fact is supported by Prop. 3, which shows that in the limit case where $\tau_1 = 0$ and $\tau_2 \to +\infty$, only one or two iterations are required for convergence (depending on the implementation).



(a) Resolut of the OT match (top row) and the Hungarian match (bottom row) on image №630



(b) Resolut of the OT match (top row) and the Hungarian match (bottom row) on image №180

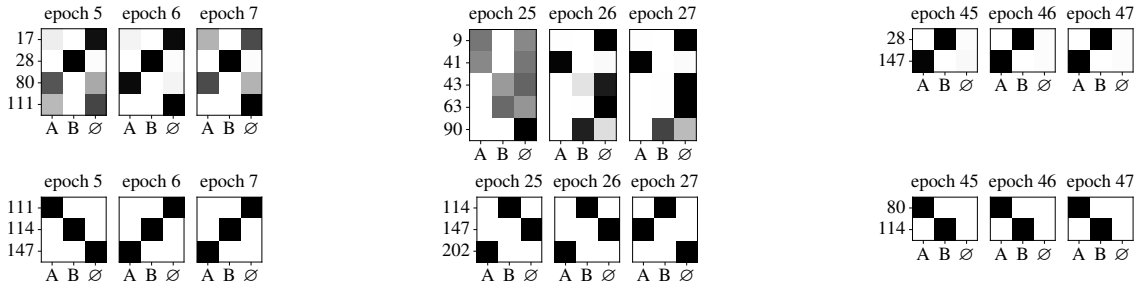


(c) Resolut of the OT match (top row) and the Hungarian match (bottom row) on image №613

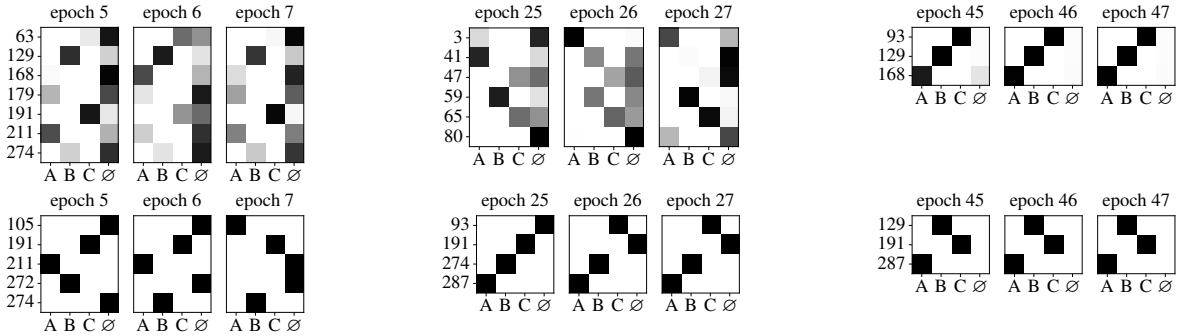
Figure 10. Output of the matching algorithms with DETR on the validation set of the Color Boxes Dataset. The model is trained two times: once with an OT match and once with a Hungarian matching. The rows indicate the predictions and the columns indicate the ground truth objects (including the background ∅). We sample examples with few ground truth objects for readability and only show predictions that are matched at least once with a non-background ground truth.

G. First Constraint Parameter

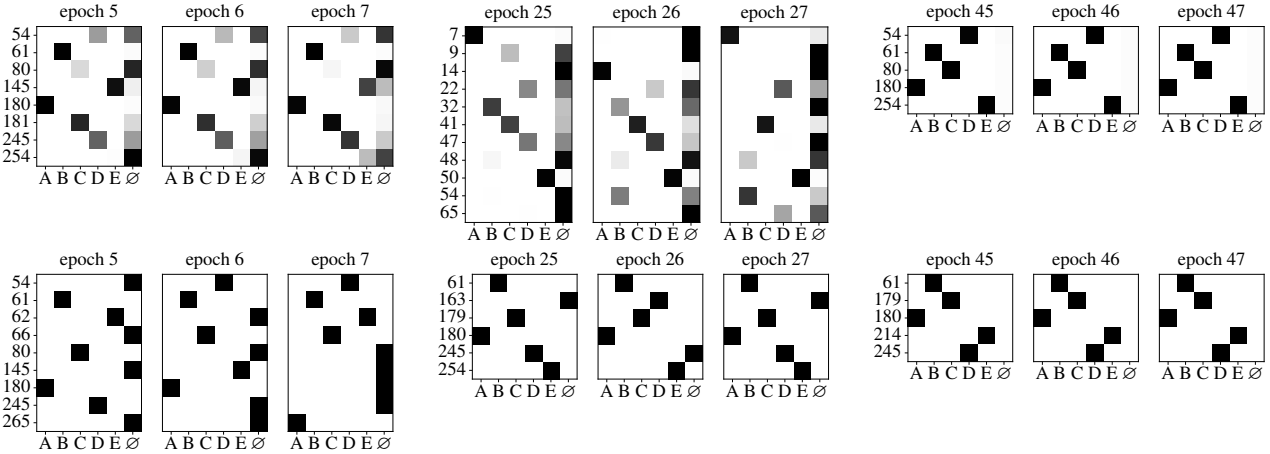
In this section, we analyze the effect of the prediction's mass constraint parameter τ_1, while we fix parameter τ_2 to a large value $\tau_2 = 100$ to simulate a hard constraint. Parameter τ_1 controls the degree to which variations in the prediction masses



(a) Resolut of the OT match (top row) and the Hungarian match (bottom row) on image №630



(b) Resolut of the OT match (top row) and the Hungarian match (bottom row) on image №180



(c) Resolut of the OT match (top row) and the Hungarian match (bottom row) on image №613

Figure 11. Output of the matching algorithms with Deformable-DETR on the validation set of the Color Boxes Dataset. The model is trained two times: once with an OT match and once with a Hungarian matching. The rows indicate the predictions and the columns indicate the ground truth objects (including the background ∅). We sample examples with few ground truth objects for readability and only show predictions that are matched at least once with a non-background ground truth.

are penalized. Each ground-truth object can be matched to the best prediction in the limit case $\tau_2 \to +\infty$ and $\tau_1 = 0$. However, some predictions cannot be matched, and others multiple times. The results for SSD on Color Boxes are displayed in Table 3. Wa can therefore conclude that the first constraint parameter τ_1 has a small influence on the metrics, both with and without NMS. Nevertheless, a higher performance is reached in the balanced case, *i.e.*, when $\tau_1 \to +\infty$.

H. Timing Analysis for SSD

As can be seen in Table 4, OT-based matches improve the epoch time (forward pass, compute the match cost, matching algorithm, and backward pass; in blue) for SSD with the Hungarian algorithm by almost 50%. The difference is smaller for DETR and variants as the models are proportionally heavier and the number of predictions smaller.

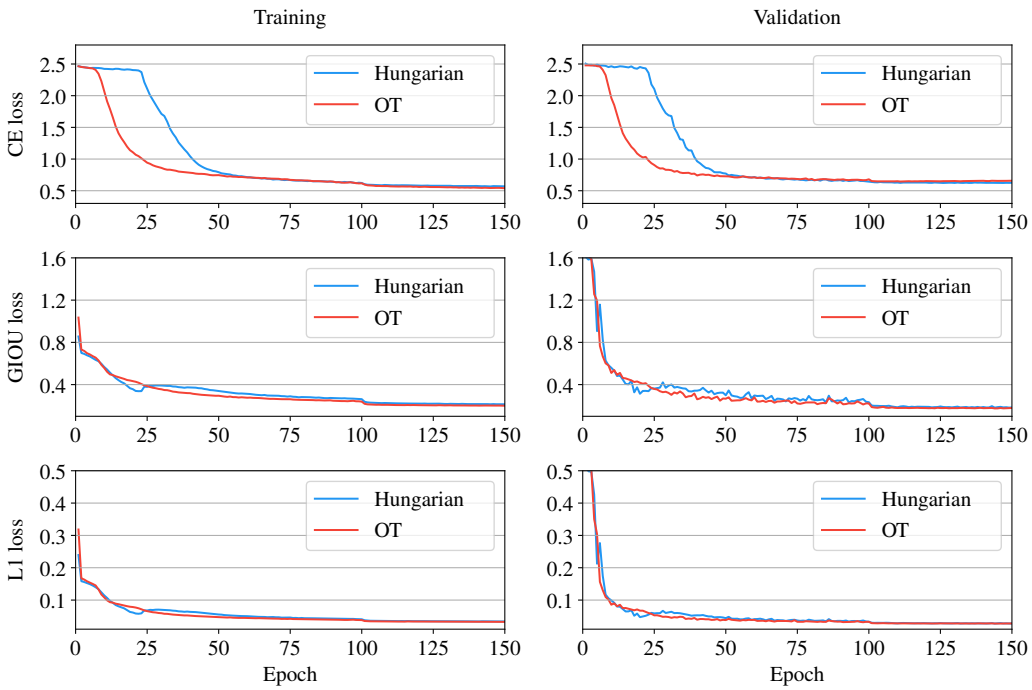


Figure 12. Training and validation unscaled loss curves for DETR on the Color Boxes dataset. The training loss is the average over the epoch.

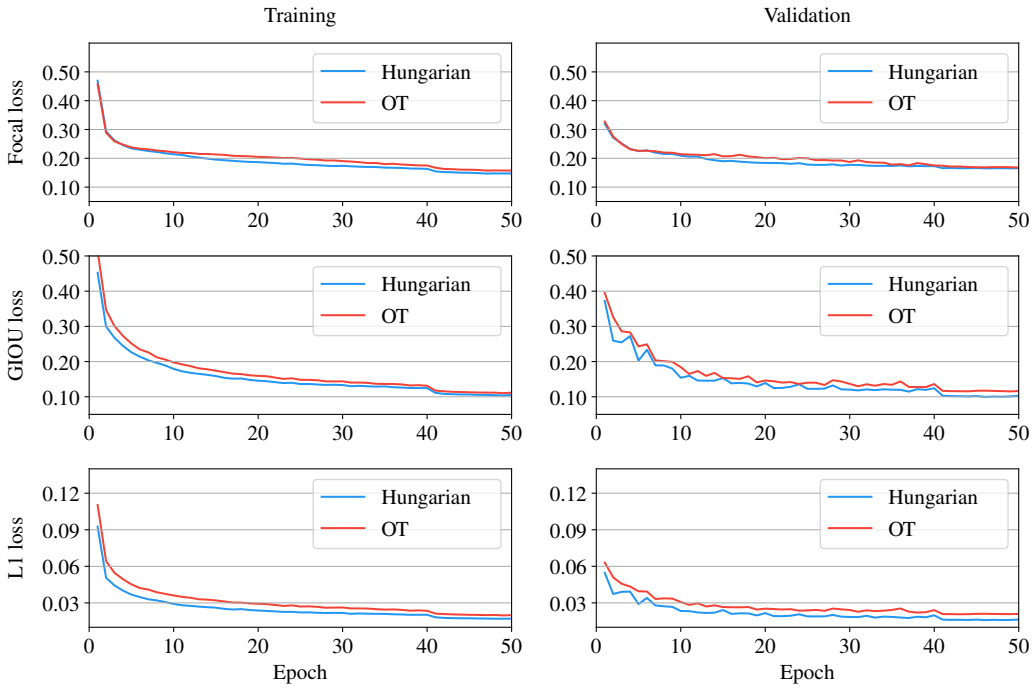
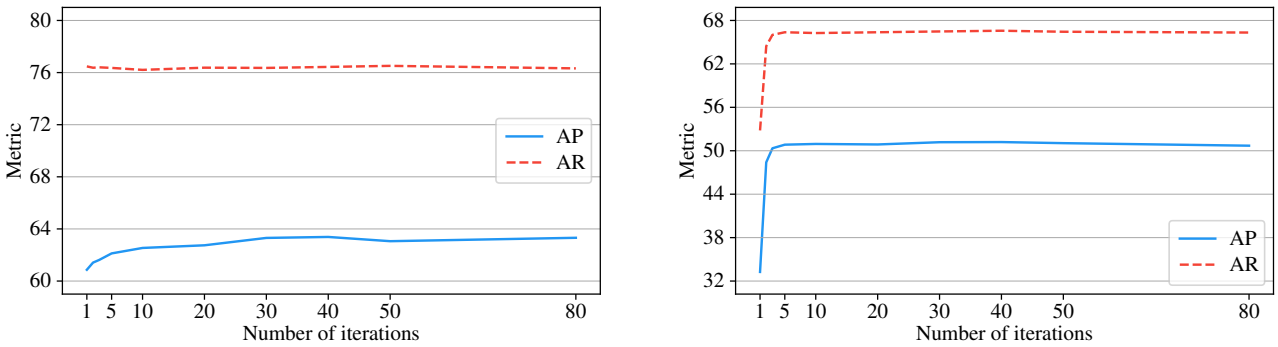


Figure 13. Training and validation unscaled loss curves for Deformable DETR on the Color Boxes dataset. The training loss is the average over the epoch.



(a) Deformable DETR with Balanced OT.

(b) SSD300 with Unbalanced OT ($\tau_2 = 0.01$).

Figure 14. Influence of the number of Sinkhorn iterations on the final metrics on the Color Boxes dataset.

Matching	τ_1	**with NMS**		**w/o NMS**	
		AP	**AR**	**AP**	**AR**
Unb. OT	0.01	47.2	62.0	41.9	71.1
Unb. OT	0.1	47.7	63.7	44.7	72.3
Unb. OT	1	47.7	64.0	44.8	72.7
Unb. OT	10	47.8	63.8	45.0	72.6
OT	(∞)	**48.1**	**64.3**	**45.2**	**73.0**

Table 3. Comparison of matching strategies on the Color Boxes dataset. SSD300 is evaluated both with and without NMS.

Epoch step	**OT**	**Unb. OT**	**Hung.**	**2-step**
Preprocessing	6.3 ms	*idem*	*idem*	*idem*
Forward pass	5.8 ms	*idem*	*idem*	*idem*
Anchor gen.	54.2 ms	*idem*	*idem*	*idem*
Match cost	4.2 ms	*idem*	*idem*	*idem*
Matching	1.1 ms	1.5 ms	18.3 ms	2.3 ms
Backward pass	8.2 ms	*idem*	*idem*	*idem*
Final losses	11.6 ms	11.6 ms	9.7 ms	9.7 ms

Table 4. Timing for each step in SSD300 on Color Boxes and a batch size of 16, computed with an Nvidia TITAN X GPU and Intel Core i7-4770K CPU @ 3.50GHz. Likewise the models we built upon, we used *Torchvision's* anchor generation implementation, which extensively relies on heavy loops and could drastically be improved (not the focus of our work). The final losses timings are partially due to the expensive hard-negative mining.

I. Color Boxes Dataset

This section provides a discussion of the Color Boxes synthetic dataset. It is split into 4,800 training and 960 validation images of 500 $\times$ 400 pixels. Images have a gray background. We uniformly randomly draw between 0 and 30 rectangles of 20 different colors, which define the category of the rectangle. The dimension of the rectangles vary from 12 to 80 pixels and are uniformly randomly rotated. They are placed such that the IoU between their bounding boxes is at most 0.25. A gaussian noise of mean 0 and standard deviation 0.05 is added to each pixel value independently. Sample images are drawn in Fig. 15.

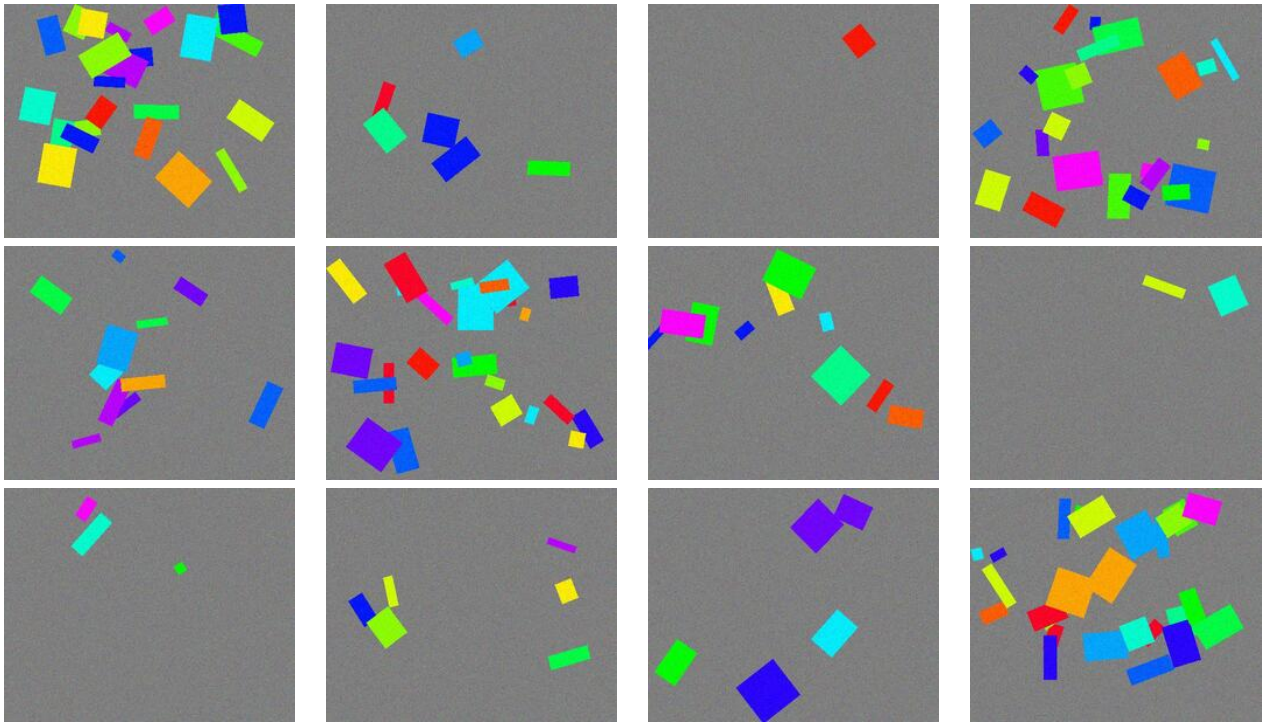


Figure 15. Sample images from the Color Boxes Dataset.

References

[1] Richard A. Brualdi. *Combinatorial Matrix Classes*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2006. 2

[2] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018. 4

[3] Lénaïc Chizat, Pierre Roussillon, Flavien Léger, François-Xavier Vialard, and Gabriel Peyré. Faster wasserstein distance estimation with the sinkhorn divergence. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. 1

[4] Aude Genevay. *Entropy-Regularized Optimal Transport for Machine Learning*. Theses, PSL University, Mar. 2019. 1

[5] Aude Genevay, Gabriel Peyre, and Marco Cuturi. Learning generative models with sinkhorn divergences. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1608–1617. PMLR, 09–11 Apr 2018. 1

[6] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 1, 4, 9

[7] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society. 1

[8] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, pages 8922–8931, 2021. 4, 5