Supplementary Material

1 CONSTRUCTION OF THE NETWORK

Here, we extensively explain the procedure used for constructing the transport network K. All the essential steps are schematically drawn in Fig. S1, and are as follows.

Step 1. Initially, a pair of images (Image 1, Image 2), is given as a couple of multidimensional arrays of dimensions $(w_i, h_i, M = 3)$, with $i = 1, 2$. We denote with w images' widths and with h their heights. The third dimension has size $M = 3$, and corresponds to the three RGB color channels. The color channels are flattened to obtain the tensors G and H, the first for Image 1, and the second for Image 2. In detail, each channel is vectorized to have dimension $m \times 1$, with $m = w_1 \cdot h_1$, for g^a (resp. $n \times 1$, with $n = w_2 \cdot h_2$, for h^a), which are inflows and outflows of our multicommodity dynamics. In this way, the tensors G and H, which are obtained stacking g^a and h^a, have size $m \times M$ and $n \times M$. Entries of G and H are in standard RGB encoding, hence they are integers ranging from 0 to 255.

To obtain the transport network K, we first generate a complete bipartite graph between $m + n = |V_1| + |V_2|$ nodes, the first $m = |V_1|$ are the pixels of Image 1, and the other $n = |V_2|$ are those of Image 2. We then assign a cost to each edge of this graph using both information given by pixels' locations, and by images' colors. In particular, we define:

$$C_e(\theta) = (1 - \theta)Y_e + \theta X_e \quad \forall e = (i, j) \tag{S1}$$

$$Y_{e=(i,j)} = ||v_i - v_j||_2 = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{S2}$$

$$X_{e=(i,j)} = ||G_i - H_j||_1 = \sum_{a=1}^{M=3} |G_{ia} - H_{ja}| \tag{S3}$$

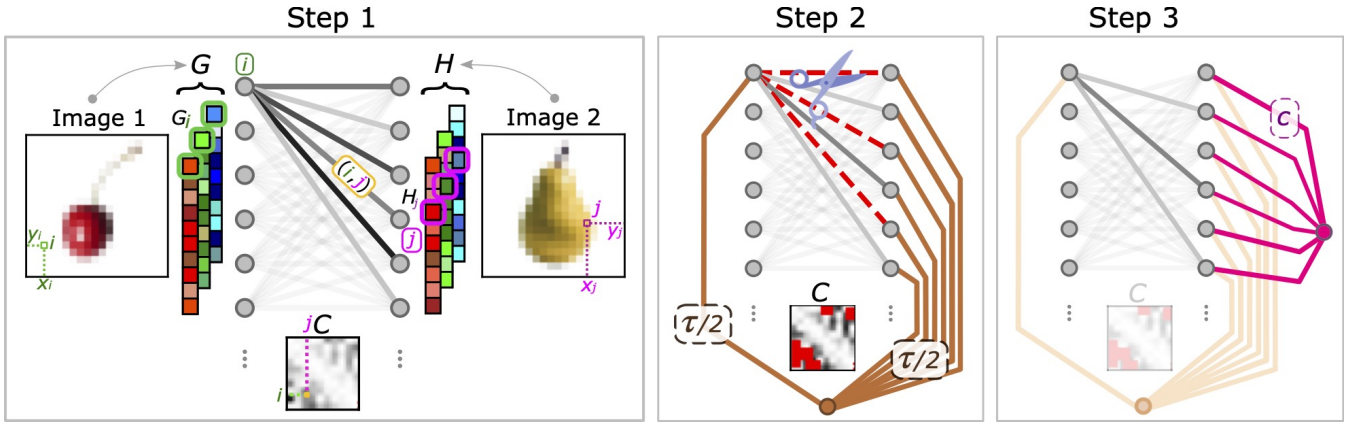

Figure S1. Detailed construction of transport networks. Step 1: conversion of colored images to tensors and construction of the first complete bipartite graph. Step 2: trimming of expensive edges and addition of the transshipment node, u_1, with its links (in brown). Step 3: relaxation of mass balance with the addition of the second auxiliary node, u_2, together with its links (in magenta).

for each i pixel of Image 1, and j pixel of Image 2. Terms Y_e in Eq. (S2) contain the Euclidean distance between any pair of pixels, whose horizontal and vertical coordinates are stored in vectors $v = (x, y)$. Instead, X_e contributes with colors to edges' costs. We model the effect of colors taking, in Eq. (S3), the 1-norm between arrays G_i, H_j, containing the RGB intensities in i (pixel of Image 1) and j (pixel of Image 2). Both X_e and Y_e have been opportunely rescaled in the range $[0, 1]$. Lastly, we use the scalar parameter $0 \leq \theta \leq 1$ to weigh Y_e and X_e in a convex combination, in Eq. (S1).

Step 2. Once Step 1 is complete, and a cost C_e is assigned to each edge of the complete bipartite graph between the two images, we implement a trimming procedure similar to that of [1, 2] to cut highly expensive links. In particular, we trim all edges e that have cost $C_e > \tau$, where $\tau > 0$ is a threshold fixed a priori. The links between V_1 and V_2 that do not get cut make up the set E_{12}. We then add a first transshipment node, u_1, to the network, and connect it with $m + n$ links to the sets V_1 and V_2. Each transshipment link is assigned a fixed cost $C_e = \tau/2$. This implies that one needs to pay a total cost of τ to transport mass from a node of Image 1 (in V_1) to one of Image 2 (in V_2), when traversing transshipment links.

There are several benefits in thresholding for the cost: (i) from a purely intuitive standpoint, humans perceive distances as saturated distances [3]; (ii) many natural color distributions are noisy and heavy-tailed, thus thresholding permits to assign a fixed cost to outliars; (iii) thresholded distances induce a W_1 distance between distributions in standard unicommodity OT problems [2]. More practically, thresholding improves accuracy and speed of OT [2] (see also the Computational Cost Section in this SM).

Step 3. The last step required to obtain K is the introduction of a second auxiliary node, u_2, together with its edges, to relax mass balance. In detail, in a standard OT setting $\sum_i G_{ia} = \sum_j H_{ja} = \Lambda^a > 0$ holds $\forall a = 1, \ldots, M$, i.e., two histograms to be transported belong to the same simplex of mass $\Lambda^a > 0$. We relax this constraint permitting $\sum_i G_{ia} \neq \sum_j H_{ja}$ and penalizing Eq. (1) (main text). Particularly, we use a similar relaxation of that in [2], which we generalize to the multicommodity setup:

$$J^\star_\Gamma(G, H) = \min_{P \in \Pi(G,H)} J_\Gamma(G, H) \quad \overset{\text{Relaxation}}{\longrightarrow} \quad \min_P \left\{ J_\Gamma(G, H) + \alpha \sum_a \left| \sum_j H_{ja} - \sum_i G_{ia} \right| \max_{e \in E_{12}} C_e \right\}. \quad \text{(S4)}$$

The intuition of Eq. (S4) is that the OT problem is penalized proportionally to the net difference between the inflowing and the outflowing mass. Hence, two images whose colors strongly differ return a higher cost and, in a supervised classification task, are less likely to be assigned the same label. We fix $\alpha = 1/2$ as in [1].

This penalization can be translated to the transport network with the addition of n links, costing $c = \alpha \max_{e \in E_{12}} C_e = \max_{e \in E_{12}} C_e/2$, connected to u_2. The excess of mass $m^a = \sum_j H_{ja} - \sum_i G_{ia}$ given by each commodity is injected u_2 to guarantee that the whole system is isolated. With this expedient one recovers exactly the relaxed OT formulation in Eq. (S4). In fact, all the transport paths that not flow into one of the n nodes of Image 2 penalize the cost by traversing the edges connected to u_2. From conservation of mass one can see that these transport paths satisfy $P^a_{ju_2} = H_{ja} - (1/n) \sum_i G_{ia}$, $\forall j \in V_2$. Thus,

summing over a and j returns exactly $\sum_{aj} P^a_{ju_2} = ||\sum_j H_j - \sum_i G_i||_1$, with the 1-norm taken over the commodities. This is precisely the penalization we introduced in Eq. (S4).

2 EQUIVALENCE BETWEEN MULTICOMMODITY DYNAMICS AND OT SETUP

With the following derivations (similar to [4, 5]), we show that asymptotic solutions of Eqs. (3)-(4) (main text) are equivalent to minimizers of Eq. (1) (main text). This implies that by solving the multicommodity dynamics we find a solution of the multicommodity OT minimization problem. More practically, for a given pair of images, running a numerical scheme on Eqs. (3)-(4) (main text) allows us to compute $\lim_{t\to\infty} P(t) = P^\star$, hence $J^\star_\Gamma = J_\Gamma|_{P=P^\star}$, and use the latter as a measure of similarity between them.

More in detail, we first demonstrate the equivalence between stationary solutions of the multicommodity dynamics and minimizers of the multicommodity OT problem introducing a second accessory minimization problem. Stationary solutions are proven to be asymptotes of Eq. (4) (main text) only afterwards, with the introduction of a Lyapunov functional for the multicommodity dynamics.

2.1 Stationary solutions of the multicomodity dynamics and OT minimizers

Initially, we observe that stationary solutions of the multicommodity dynamics satisfy the relation

$$x_e = ||P_e||_2^{2/(1+\gamma)} \quad \forall e \in E, \tag{S5}$$

that one can derive setting the left hand side of Eq. (4) (main text) to zero, defining $P^a_e = x_e(\phi^a_i - \phi^a_j)/C_e$ for $e = (i, j)$, and $\gamma = 2 - \beta$. We recover an scaling identical to Eq. (S5) introducing the following auxiliary constrained minimization problem:

$$\min_{x,P} \left\{ \frac{1}{2} \sum_e \frac{C_e}{x_e} ||P_e||_2^2 + \frac{1}{2\gamma} \sum_e C_e x_e^\gamma \right\} \tag{S6}$$

$$\text{s.t. } \sum_e B_{ie} P^a_e = S^a_i \quad \forall i \in V, a = 1, \dots, M. \tag{S7}$$

In fact, differentiating with respect to x_e the objective function in Eq. (S6), and setting the derivatives to zero, yields

$$-\frac{C_e}{x_e^2} ||P_e||_2^2 + C_e x_e^{\gamma-1} \stackrel{!}{=} 0 \quad \longrightarrow \quad x_e = ||P_e||_2^{2/(1+\gamma)} \quad \forall e \in E. \tag{S8}$$

Noticeably, Eq. (S6) admits a straightforward physical interpretation. In fact, the first term $J = (1/2)\sum_e C_e ||P_e||_2^2/x_e$ is Joule's first law. Particularly, transport paths can be thought of as fluxes of mass transported through the edges of a capacitated network with resistances $r_e = C_e/x_e$. While the second term, $W_\gamma = (1/2\gamma)\sum_e C_e x_e^\gamma$, is the cost needed to build the network infrastructure. The constraints in Eq. (S7)—identical to Eq. (3) (main text)—are equivalent to Kirchhoff's law, enforcing conservation of mass.

Most remarkably, the scaling of Eq. (S8) can be also recasted in Eq. (S6) to find that $J_\Gamma = J + W_\gamma$ (neglecting multiplicative constants). This connects the multicommodity dynamics with the objective

function of Eq. (1) (main text). In detail,

$$J + W_\gamma = \frac{1}{2}\sum_e \frac{C_e}{x_e}||P_e||_2^2 + \frac{1}{2\gamma}\sum_e C_e x_e^\gamma \tag{S9}$$

$$\overset{\text{Eq. (S8)}}{=} \frac{1}{2}\sum_e C_e||P_e||_2^{2\gamma/(1+\gamma)} + \frac{1}{2\gamma}\sum_e C_e||P_e||_2^{2\gamma/(1+\gamma)} \tag{S10}$$

$$\overset{\Gamma=2\gamma/(1+\gamma)}{=} \frac{1}{\Gamma}\sum_e C_e||P_e||_2^\Gamma \tag{S11}$$

$$= \frac{1}{\Gamma}J_\Gamma(G,H). \tag{S12}$$

To complete the mapping between the multicommodity dynamics and the minization setup, we show that the space of transport tensors $\Pi(G,H)$ is exactly the same space defined by Eq. (S7). This can be seen with the following chain of equalities:

$$\sum_k P_{ik}^a - \sum_j P_{ji}^a = G_i^a - H_i^a \quad \forall i \in V, a = 1,\dots,M \tag{S13}$$

$$\sum_k P_{e=(i,k)}^a - \sum_j P_{e=(j,i)}^a = S_i^a \quad \forall i \in V, a = 1,\dots,M \tag{S14}$$

$$\sum_e B_{ie}P_e^a = S_i^a \quad \forall i \in V, a = 1,\dots,M. \tag{S15}$$

Here we take the difference between the OT constraints of $\Pi(G,H)$ in Eq. (S13), we then use the definition of S in Eq. (S14), and compact the plus and minus signs using the signed incidence matrix B in Eq. (S15). This allows us to recover Kirchhoff's law as formulated in Eq. (S7) and Eq. (3) (main text).

2.2 Multicommodity dynamics asymptotes: Lyapunov functional

We complete our discussion introducing the Lyapunov functional for Eq. (4) (main text) proposed in [4, 5]. The functional reads:

$$\mathcal{L}_\gamma[x] = \frac{1}{2}\sum_{ai}\phi_i^a[x]S_i^a + \frac{1}{2\gamma}\sum_e C_e x_e^\gamma, \tag{S16}$$

and it is a multicommodity generalization of that originally introduced in [6]. This is a well-defined Lyapunov functional for the multicommodity dynamics, in fact, along a curve $x(t)$ solution of Eq. (4) (main text),

$$\frac{d\mathcal{L}_\gamma[x(t)]}{dt} \leq 0. \tag{S17}$$

With the equality satisfied if and only if $x(t)$ is a stationary point of Eq. (4) (main text). This can be shown as follows. We claim that

$$\frac{\partial\mathcal{L}_\gamma}{\partial x_e} = \frac{C_e}{2}\left(x_e^{\gamma-1} - \frac{||\phi_i - \phi_j||_2^2}{C_e^2}\right) \quad \forall e = (i,j) \in E. \tag{S18}$$

This equality can be retrieved differentiating both sides of Eq. (3) (main text) by x_e, thus obtaining

$$\sum_j \frac{\partial L_{ij}}{\partial x_e}\phi_j^a + \sum_j L_{ij}\frac{\partial \phi_j^a}{\partial x_e} = 0 \quad \forall i \in V, e \in E, a = 1, \dots, M, \tag{S19}$$

$$\sum_j L_{ij}\frac{\partial \phi_j^a}{\partial x_e} = -\sum_j B_{je}(1/C_e)B_{ie}\phi_j^a \quad \forall i \in V, e \in E, a = 1, \dots, M. \tag{S20}$$

Then, multiplying Eq. (S20) by ϕ_i^a and summing over i one gets

$$\sum_{ij} \phi_i^a L_{ij}\frac{\partial \phi_j^a}{\partial x_e} = -\sum_{ij} \phi_i^a B_{ie}(1/C_e)B_{je}\phi_j^a \quad \forall e \in E, a = 1, \dots, M, \tag{S21}$$

further summing over a yields

$$\frac{\partial}{\partial x_e}\left(\sum_{aj} S_j^a \phi_j^a\right) = -C_e\frac{||\phi_i - \phi_j||_2^2}{C_e^2} \quad \forall e = (i,j) \in E, \tag{S22}$$

where in the left hand side of Eq. (S22) we used Eq. (3) (main text). From Eq. (S22) the equality in Eq. (S18) follows immediately. Now, thanks to Eq. (S18) we can prove that the Lie derivative of the functional is less than or equal to zero. In fact,

$$\frac{d\mathcal{L}_\gamma}{dt} = \sum_e \frac{\partial \mathcal{L}_\gamma}{\partial x_e}\frac{dx_e}{dt} \tag{S23}$$

$$\overset{\text{Eq. (S18)}}{=} \sum_e \frac{C_e}{2}\left(x_e^{\gamma-1} - \frac{||\phi_i - \phi_j||_2^2}{C_e^2}\right)\frac{dx_e}{dt} \tag{S24}$$

$$\overset{\text{Eq. (4)},\gamma=2-\beta}{=} -\sum_e \frac{C_e}{2}x_e^{2-\gamma}\left(x_e^{\gamma-1} - \frac{||\phi_i - \phi_j||_2^2}{C_e^2}\right)^2 \leq 0. \tag{S25}$$

With the equality in Eq. (S25) that is recovered if and only if (i) $x_e = 0$, or (ii) the scaling in Eq. (S8) holds.

Finally, we show that the Lyapunov is identical to the total sum of dissipation and transport cost, i.e., $\mathcal{L}_\gamma = J + W_\gamma$. This can be done multiplying both sides of Eq. (3) (main text) by ϕ_i^a and then summing over i and a, namely

$$\sum_{aiej} \phi_i^a B_{ie}(x_e/C_e)B_{je}\phi_j^a = \sum_{ai} \phi_i^a S_i^a \tag{S26}$$

$$\sum_e \frac{C_e}{x_e}||P_e||_2^2 = \sum_{ai} \phi_i^a S_i^a \tag{S27}$$

where we used $P_e^a = x_e(\phi_i^a - \phi_j^a)/C_e$, for $e = (i,j)$. This allows us to conclude.

In summary, we showed that the multicommodity dynamics admits a well-defined Lyapunov functional, which is equivalent to the sum of a dissipation and an infrastructure cost. These two contributions, which are jointly minimized by Eq. (4) (main text), when evaluated along their minimizers correspond to the multicommodity OT cost J_Γ of Eq. (1) (main text). Introducing the

Lyapunov functional is crucial to formally show that asymptotics of the dynamics are equivalent to minimizers of the cost, namely $\lim_{t\to\infty} P(t) = P^\star$.

Lastly, we remark the effect of γ (resp. β) on the minimization problem. In the setting where $\gamma > 1$ ($\beta < 1$) the functional $\mathcal{L}_\gamma$ is convex, with one unique minimizer. For $\gamma < 1$ ($\beta > 1$) the functional landscape becomes rugged and strongly non-convex, with multiple minimizers each correspondent to a local minima of the cost. Hence, in this second scenario, running Eq. (4) (main text) permits to converge in a stationary point, which however may not be its global minimum.

3 CROSS-VALIDATION: FLOWERS DATASET

We perform a 4-fold cross validation on both parameters used for the construction of the ground cost, θ and τ, and on algorithms' regularization parameters, β and ε. We briefly summarize it in this section.

The JF30 Dataset [7] is made of 1,479 elements, divided in 30 classes. First, we separate it into two subsets: *train* and *test*, with classes' frequencies being the same in these subsets as in the entire dataset. To cross-validate our methods, we further separate the train set into 4 folds of equal size, each to be used in turn as a validation set. More in detail, each experiment is executed fixing the validation fold and an image belonging to it, then, the Optimal Transport costs $J^\star_\Gamma$ between such image all the other images in the train set—made of the other three folds—is calculated. This procedure is repeated for all images in the validation set, and swapping each of the 4 train folds as validation set. We use a k-nearest neighbors classifier over $J^\star_\Gamma$ to assign to an image in the validation set its label, that is, for each validation image we consider the k train samples with lowest $J^\star_\Gamma$, and label the validation sample with the most frequent class among these k. This allows us to calculate the classification accuracy of a given fold, and then to average the accuracy over the 4 permutations of the validation and train set. The total amount of experiments we ran in order to cross-validate the model is approximately 50,000.

Results are shown in Fig. S2 and Fig. S3. These depict the average accuracy of: (A) the multicommodity ($M = 3$) dynamics; (B) the unicommodity ($M = 1$) dynamics, both for $\beta \in \{0.5, 0.75, 1, 1.25, 1.5\}$; (C) Sinkhorn algorithm on colored images (Sinkhorn RGB); and (D) Sinkhorn algorithm on grayscale images (Sinkhorn GS), for $\varepsilon \in \{100, 250, 500, 750, 1000, 2500\}$. Letters in parentheses refer to those of Fig. S2 and Fig. S3. The regularization parameters are validated together with $\tau \in \{0.1, 0.125\}$ and $\theta \in \{0, 0.25, 0.5, 0.75\}$. Both multicommodity and unicommodity dynamics are have initial conditions $x_e(0) = 1, \forall e \in E$.

All figures displayed in Fig. S2 and Fig. S3 correspond to highest accuracies returned by the k-NN classifier, with $k = 1, 2, \ldots, 20$. Observing the results, one can see that best performances are attained at $(\tau, \theta, \beta) = (0.125, 0.25, 1)$ for the multicommodity dynamics, and at $(\tau, \theta, \beta) = (0.125, 0.25, 1.25)$ for the unicommodity dynamics.

Noticeably, the accuracy monotonically increases (resp. decreases) with β for a fixed value of θ, namely $\theta = 0$ (resp. $\theta = 0.25$). This can be addressed to the fact that, when no color information is taken into account in the construction of the ground metric ($\theta = 0$), it is more advantageous to consolidate transport paths on cheap edges correspondent to pixels whose positions are close, thus choosing a larger β. On the other hand, introducing colors in C ($\theta > 0$), and thus creating a more disordered ground cost matrix, favors distributing transport paths on the network (See Model Interpretability Section in this SM).

Remarkably, τ also has an impact on the classification accuracy of our algorithms: the larger we set its value to be—thus trimming less edges from the transport network—the more accurate the classification becomes. This behavior is evidently different for Sinkhorn algorithm, as explained here below.

Cross-validation of Sinkhorn algorithm is taken a step further. Motivated by the classification accuracy drop observed in Fig. S2, Fig. S3 [(C), (D)] when enlarging the trimming threshold from $\tau = 0.1$ to $\tau = 0.125$, we fix θ and ε to the best values in Fig. S2 [(C), (D)], and progressively reduce τ. Results are shown in Fig. S4 (A) for Sinkhorn on grayscale images, and in Fig. S4 (B) for Sinkhorn on colored images.

Notice that both Sinkhorn GS and Sinkhorn RGB returns bell-shaped curves when changing τ. In particular, low classification accuracy is attained when strongly reducing τ, as well as when the trimming threshold is high (approximately $\tau \geq 0.5$). In the first case, many elements of the ground cost matrix are cut, and not enough information is taken into account into the OT setup to properly perform classification. In the second, too much noise is included in into C, which also negatively affects classification. More in detail, we observe in Fig. S4 (a, inset), that Sinkhorn GS performs

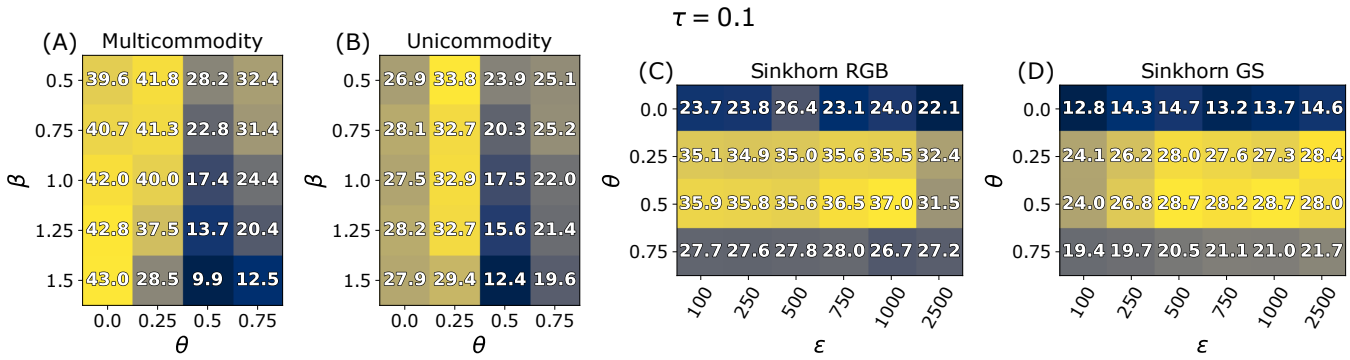

Figure S2. Cross-validation results for $\tau = 0.1$. Figures are accuracy values obtained with the 4-fold cross validation on JF30. Cells are colored with a darkest-to-brightest palette based on the accuracies. Subplots correspond to: (**A**) the multicommodity dynamics, (**B**) the unicommodity dynamics, (**C**) Sinkhorn on colored images, and (**D**) Sinkhorn on grayscale images.

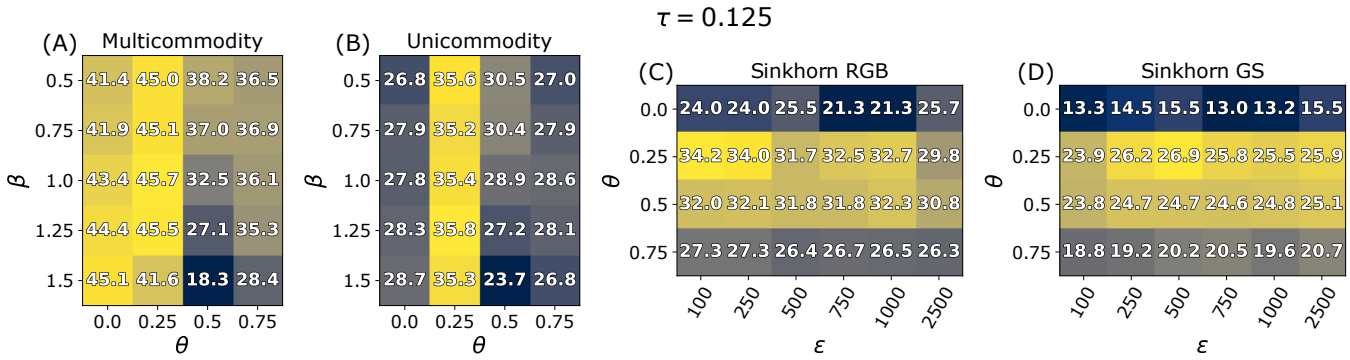

Figure S3. Cross-validation results for $\tau = 0.125$. Figures are accuracy values obtained with the 4-fold cross validation on JF30. Cells are colored with a darkest-to-brightest palette based on the accuracies. Subplots correspond to: (**A**) the multicommodity dynamics, (**B**) the unicommodity dynamics, (**C**) Sinkhorn on colored images, and (**D**) Sinkhorn on grayscale images.

best when $\tau = 0.05$. For Sinkhorn RGB, i.e. Fig. S4 (b, inset), there is a plateau for all values of the threshold within the interval $[0.05, 0.1]$.

These observations lead us to the choice of $\tau = 0.05$ for Sinkhorn GS, that we re-cross-validate ranging $\theta \in \{0.25, 0.5\}$ and $\varepsilon \in \{100, 250, 500, 750, 1000, 2500\}$. Looking at the results in Fig. S5, we note that optimal parameters for Sinkhorn GS are $(\theta, \varepsilon) = (0.25, 500)$ and $(\theta, \varepsilon) = (0.5, 500)$, which return identical classification accuracy.

As for Sinkhorn RGB, we fix the trimming threshold at the two ends of the plateau in Fig. S4 (b, inset), $\tau = 0.05$ and $\tau = 0.1$, and re-cross-validate them with $(\theta, \varepsilon) = (0.25, 100)$ and $(\theta, \varepsilon) = (0.5, 1000)$. Here, we choose two disparate values of θ and ε not being able to observe a clear relation between these two variables in Fig. S2 (C) and Fig. S3 (C). Namely, $\varepsilon = 100$ (low) and $\theta = 0.25$ perform better for $\tau = 0.125$, in contrast to $\varepsilon = 1000$ (high) and $\theta = 0.5$ for $\tau = 0.1$. Results are in Table S1, optimal parameters are $(\theta, \tau, \varepsilon) = (0.25, 0.05, 100)$.

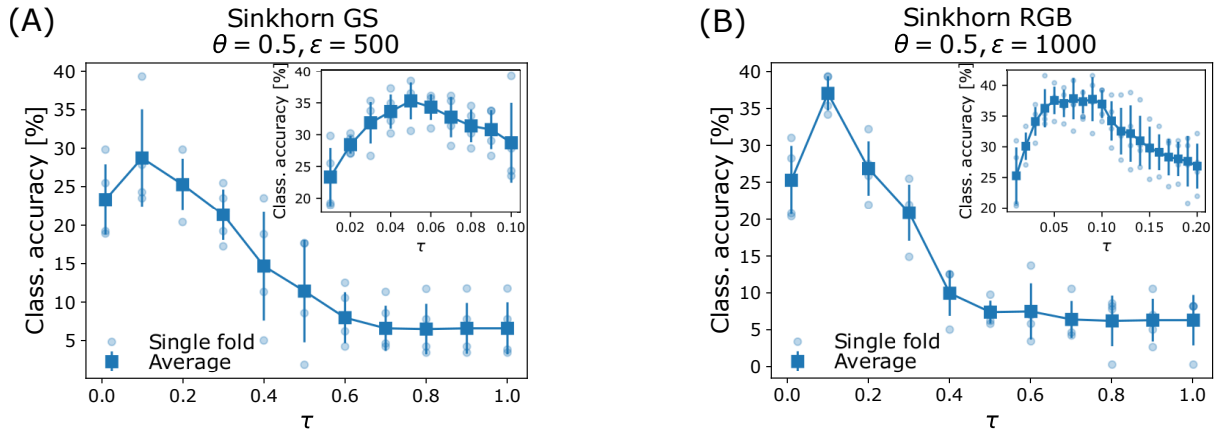

Figure S4. Sinkohrn's cross-validation results varying τ. Subplots correspond to: (**A**) Sinkhorn GS, (**B**) Sinhorn RGB. In each subplot, circular markers correspond to the accuracy values of each fold, instead squares and bars represent their average and standard deviations. In the insets, we refined the grid of τ in an interval of interest, where classification accuracy is peaked.

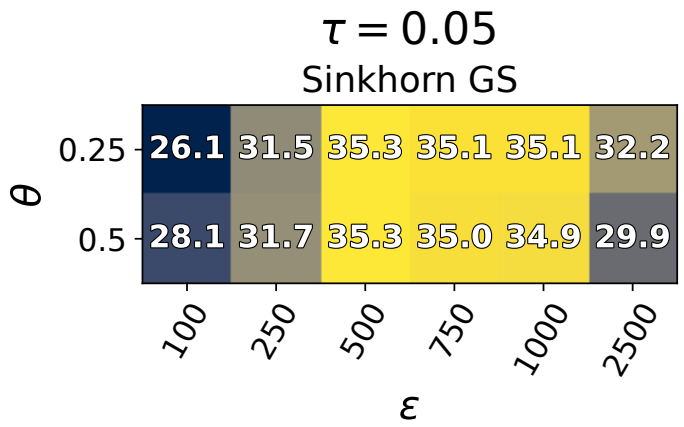

Figure S5. Refined cross-validation results Sinkhorn GS and $\tau = 0.05$. Figures are accuracy values obtained with the 4-fold cross validation on JF30. Cells are colored with a darkest-to-brightest palette based on the accuracies.

Algorithm	Hyperparameters				Class. accuracy
	θ	τ	ε	k	[%] (↑)
Sinkhorn RGB	0.25	0.05	100	1	58.4
	0.5	0.05	1000	1	53.6
	0.25	0.1	100	1	53.2
	0.5	0.1	1000	1	49.0

Table S1. Refined cross-validation results for Sinkhorn RGB. Rows are sorted (from bottom to top) using the average percent accuracy obtained with the 4-fold validation (from worst to to best).

4 EXPERIMENTAL DETAILS: FRUITS DATASET

Here, we describe in detail the experimental setup designed for the Fruit Dataset (FD) [8]. FD consists of 163 images of 15 fruit types. We split the whole dataset into *train* and *test* sets, each with 70% and 30% of the available images, respectively. As for the other dataset, classes' frequencies are the same in these subsets as in the entire dataset. Given the rather small size of this dataset, we directly perform classification comparing train and test. All the experiments have been executed with the two best performing parameter configurations of ε and θ, cross-validated on JF30, for Sinkhorn-based methods. These are: $(\theta, \varepsilon) = (0.25, 500)$, $(\theta, \varepsilon) = (0.5, 500)$ for Sinkhorn GS [see Fig. S2 (D)], and $(\theta, \varepsilon) = (0.5, 1000)$, $(\theta, \varepsilon) = (0.5, 750)$ for Sinkhorn RGB [see Fig. S2 (C)]. For our dynamics, we selected the two best performing values of β, for $\theta = 0$ and $\theta = 0.25$. Namely, $(\theta, \beta) = (0.25, 1)$, $(\theta, \beta) = (0.25, 1.5)$ for the multicommodity dynamics [see Fig. S3 (A)], and $(\theta, \beta) = (0.25, 1.25)$, $(\theta, \beta) = (0, 1.5)$ for the unicommodity dynamics [see Fig. S3 (B)]. The trimming threshold is ranged in $\tau \in \{0.04, 0.05, 0.06, 0.07\}$.

5 IMAGE PREPROCESSING

The elements of both datasets are processed in the following way. First, each image is coarsened with an average pooling, the only input needed for this step is the size of the square mask, `ms`. Its stride is in fact set to `stride` = `ms`, and the padding to `pad` = 0. All images were conveniently trimmed so that both their widths and heights are divisible by the pooling mask size. We set `ms` = 40 for JF30, and `ms` = 30 for FD. Furthermore, we smooth the images using a Gaussian filter on each color channel, with standard deviation $\sigma = 0.5$.

Moreover, to convert colored images into grayscale ones, which are given as input to Sinkhorn GS and to our unicommodity dynamics ($M = 1$), we preproces them as follows. Let (R, G, B), be the three color channels composing each pixel of a colored image, these are converted into a unique channel (its grayscale counterpart), whose intensity I is calculated with the weighted sum $I = 0.2125R + 0.7154G + 0.0721B$. The weights correspond to those used by cathode-ray tube (CRT) phosphors as they are more suitable to represent human perception of red, green and blue than equally valued weights [9].

5.1 Color distributions of images

As shown in Table I (main text), for the multicommodity and the unicommodity dynamics, optimal values of the trimming threshold τ are much lower in the Fruit Dataset [8] than in the Jena Flowers 30 Dataset [7]. This can be addressed to the fact that color distributions of fruits, belonging to the first dataset, are drastically light-tailed compared to those of flowers in the second dataset. Thus,

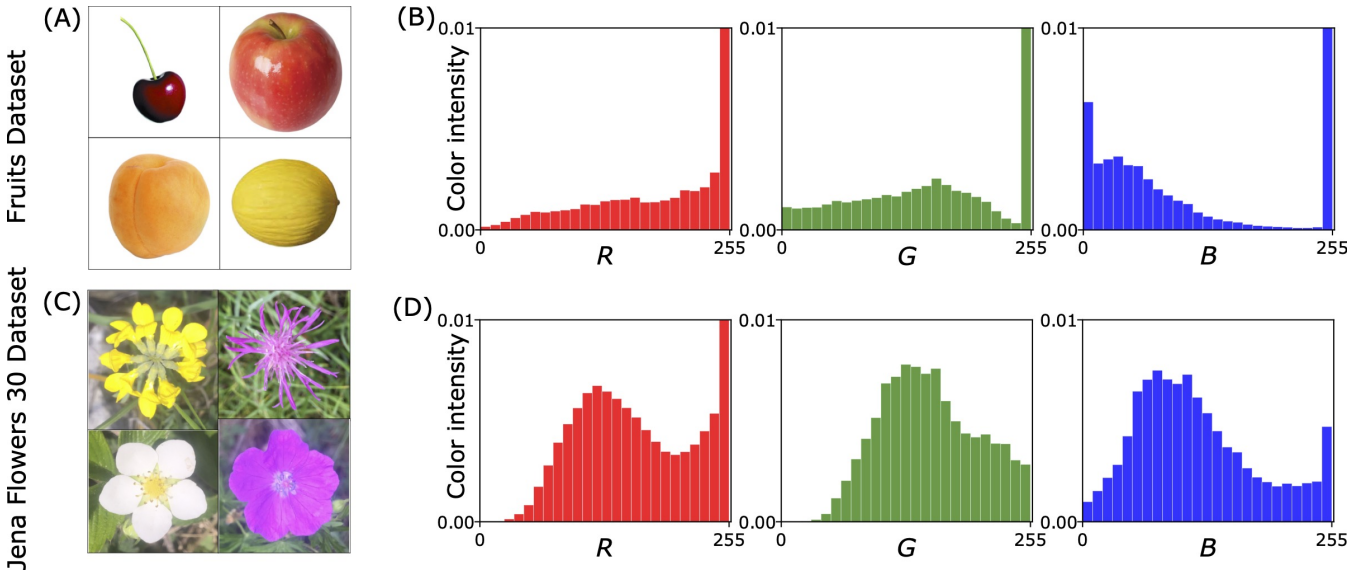

Figure S6. Color distributions in the two datasets. Subplots (**A**), (**B**) are relative to FD, subplots (**C**), (**D**) to JF30. In (A), (C) we plot four random images drawn from the two datasets. In (B), (D) the average color intensities (properly normalized to sum to one) of 100 random samples extracted from the two datasets. The plots correspond to red $= R$, green $= G$, and blue $= B$.

the cost C is naturally noisier in the latter case, and a larger trimming is necessary to remove such noise from classification.

Most of the noise in pictures of flowers comes from the background. In fact, while all flowers are photographed in nature, fruits are depicted on a white background. This can be seen in Fig. S6 (A)-(D). In subplot (A) we show four images randomly sampled from the Fruit Dataset, in (C) four random samples of the Jena Flowers 30 Dataset. In (B) and (D) we plot the average color intensity of the RGB channels for 100 random samples belonging to the two datasets. Here, the histograms in (B) are relative to the fruits, those in (D) to the flowers. From the plots it can be clearly seen that the color distributions of Fig. S6 (B) are starkly peaked around $(R, G, B) = (255, 255, 255) =$ white in standard RGB encoding.

6 SINKHORN BENCHMARKS

In our experiments, we compare the multicommodity and unicommodity dynamics against Sinkhorn algorithm, popularized by the seminal work of [10]. The idea of Sinkhorn is to regularize the standard OT problem by adding an entropic barrier to the cost function. More in detail, and following the notation adopted in our manuscript, the minimization problem proposed in [10] is:

$$P \,\text{s.t.}\, \min_{\substack{\sum_j P_{ij}=g_i \\ \sum_i P_{ij}=h_j}} \left\{ \sum_{ij} P_{ij} C_{ij} - \varepsilon h(P) \right\}, \qquad h(P) = -\sum_{ij} P_{ij} \log P_{ij}. \tag{S28}$$

Here transport paths P, which generally lie in the polyhedral set described by the constraints $\sum_j P_{ij} = g_i \,\forall i$ and $\sum_i P_{ij} = h_j \,\forall j$, are smoothed by the entropy $h(P)$. This trick makes the

optimization problem strictly convex, and permits to solve it with a very efficient matrix scaling algorithm—Sinkhorn's fixed point iteration.

We generalize the problem in Eq. (S28) in order to take in account transport tensors, G and H, which carry information of multiple color channels, and transport paths P. In detail, we propose the following minimization problem for each commodity—color channel—a,

$$P^a \text{ s.t. } \min_{\substack{\sum_j P^a_{ij}=G^a_i \\ \sum_i P^a_{ij}=H^a_j}} \left\{ J^a_{\text{sink}} = \sum_{ij} P^a_{ij} C_{ij} - \varepsilon h(P^a) \right\}, \qquad h(P^a) = -\sum_{ij} P^a_{ij} \log P^a_{ij}. \tag{S29}$$

This allows to efficiently compute, using Sinkhorn's scaling, an Optimal Transport path P^a_{opt} for each commodity, together with its correspondent optimal cost $J^a_{\text{sink,opt}} = J^a_{\text{sink}}|_{P^a=P^a_{\text{opt}}}$. Finally, the Optimal Transport cost for colored images is calculated as $J^{\text{RGB}}_{\text{sink,opt}} = (1/3)\sum_{a=1}^{M=3} J^a_{\text{sink}}$.

7 MODEL INTERPRETABILITY

In this section we discuss the effect that the parameters θ, τ, and β have one the OT setup.

First, we explain the experiment in Fig. S7. We start by sampling two images of the FD dataset belonging to the same class. These images have identical shape, i.e. width w and height h equal to 20. They are displayed on the leftmost part of Fig. S7. From these two images, we obtain the tensors G and H, that are transported in the OT problem. The first, G, is constructed using all the pixels on the 11th row of Image 1, thus its dimension is $m \times M = 20 \times 3$. The same row of Image 2 is used to build H, also in this case its size is $n \times M = 20 \times 3$.

The two tensors enter in Eq. (1) (main text) together with a (20×20)-dimensional cost C, which is built with pixels' locations and color information using Eqs. (S1)-(S3). The scope of this discussion is to refine the intuition on these formulas, and on the effect that θ and τ have on C. In Fig. S7 we plot the ground cost C for the two tensors G and H, for $\theta = \{0, 0.25, 0.5, 0.75\}$ and $\tau = \{0.1, 0.25, 0.5, 1\}$. All entries $C_{ij} > \tau$—which correspond to those edges that are trimmed from the transport network—are colored in white.

Notice that for $\theta = 0$ all costs are symmetric. Indeed in this case $C_{ij} = \min\{Y_{ij}, \tau\}$, with Y that is containing the Euclidean distances between pixels' coordinates, i.e. Eq. (S2). Here, decreasing the trimming threshold τ progressively sparsifies the banded matrices drawn in the first column of Fig. S7, with limit cases being $C = \text{diag}[C_{0,0}, C_{1,1}, \ldots, C_{19,19}]$—for τ sufficiently small, and $C = Y$—for $\tau \geq \max_{ij} Y_{ij}$. On the other hand, the symmetry is gradually broken as θ is increased, namely, when colors of the images are are used to build into C. This is clearly depicted in Fig. S7, where the heatmaps get progressively disordered for larger values of θ (from left to right).

To further expand this discussion, we design a second experiment, schematically represented in Fig. S8. Here, we solve the OT problem between two tensors, G and H, built similarly to those of Fig. S7. Particularly, we consider three central pixels of the 11th rows of Image 1 and Image 2, as drawn in the leftmost part of the Figure, so that both g^a and h^a are 3-dimensional arrays for all $a = 1, \ldots, 3$, and C is a (3×3)-dimensional matrix.

Depending on the values of θ, the ground cost C is either symmetric ($\theta = 0$), and computed only using pixels' coordinates, or strongly irregular ($\theta = 0.75$), since colors of images are taken into

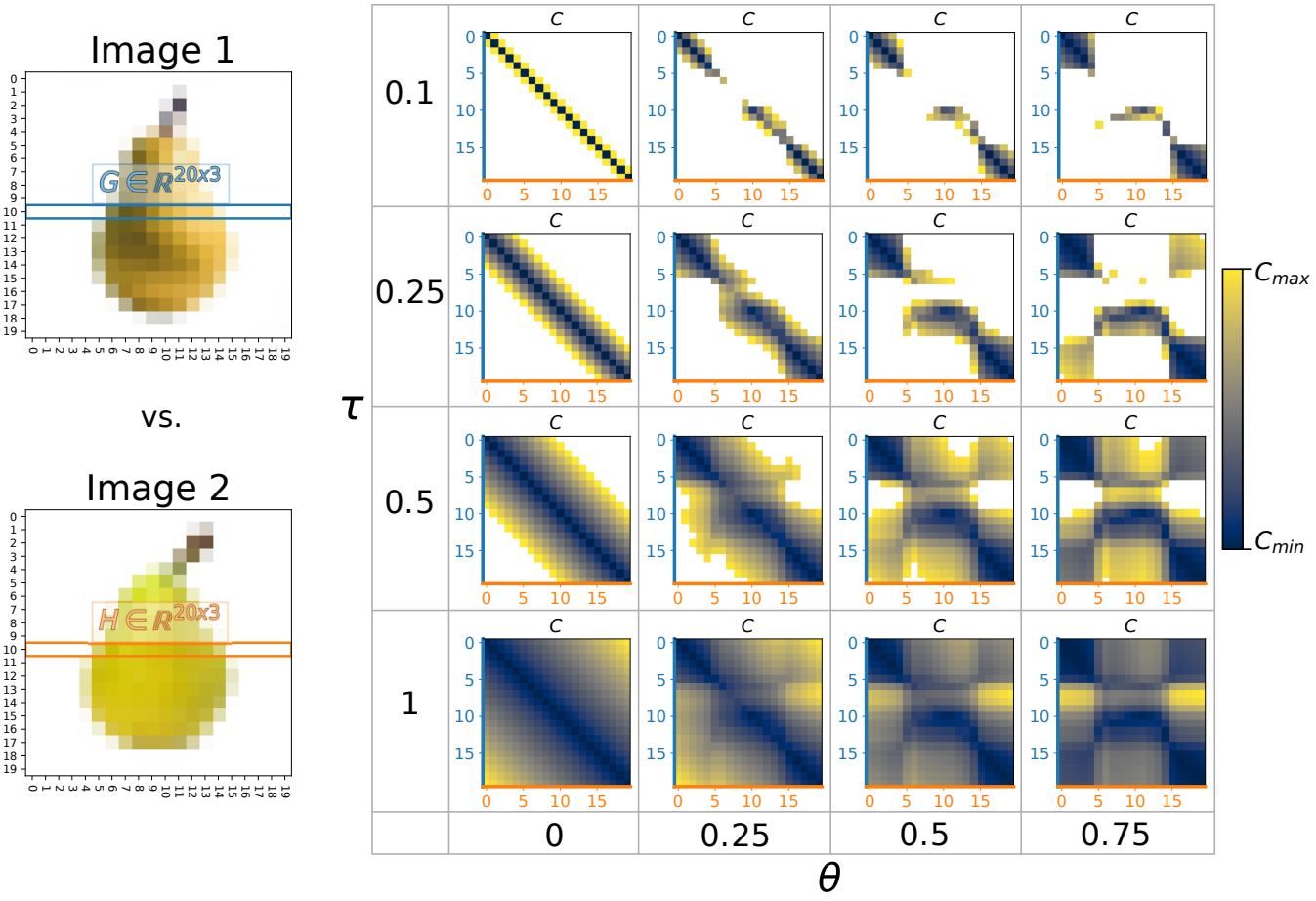

Figure S7. Effect of θ and τ on OT. On the left, we display the two samples used to build the ground costs C. Highlighted rows in blue and orange are those considered to extract G and H. On the right side of the panel we plot C for $\theta = \{0, 0.25, 0.5, 0.75\}$ and $\tau = \{0.1, 0.25, 0.5, 1\}$. White regions correspond to trimmed values, i.e. entries of C that are larger than τ.

account. In the first case, the transport network connecting the images has also a symmetric structure. Here, elements along the diagonal of the cost—correspondent to horizontal edges connecting orange and blue nodes with the same index—are much cheaper than all the other entries. This is due to the fact that the Euclidean distance between two pixels with the same position is zero (practically set to a safety default value $\epsilon = 10^{-5}$). Conversely, in the second case, introducing colors in the ground cost translates into having higher values along the diagonal elements of C. Here, colors—which distribute more smoothly on images—smooth out the cost as well, whose entries are more homogeneous.

As shown in Fig. S2 and Fig. S3, taking a purely Euclidean ground cost C, i.e., $\theta = 0$, returns higher classification accuracy when $\beta = 1.5$. Instead, building C mostly with color information, thus setting $\theta = 0.75$, favors $\beta = 0.5$. We address this tendency to the effect that β has on transport paths' consolidation, and we represent it on the rightmost portion of Fig. S8, where we plot the Optimal Transport paths $\{P^1, P^2, P^3\}$ obtained running Eqs. (3)-(4) (main text) on the OT setup just discussed. In detail, for $\theta = 0$, horizontal edges in the transport network are much cheaper than the others, therefore strong consolidation of transport paths ($\beta = 1.5$) benefits classification. Conversely, since for $\theta = 0.75$ the entries of C are more homogeneous, distributing transport paths

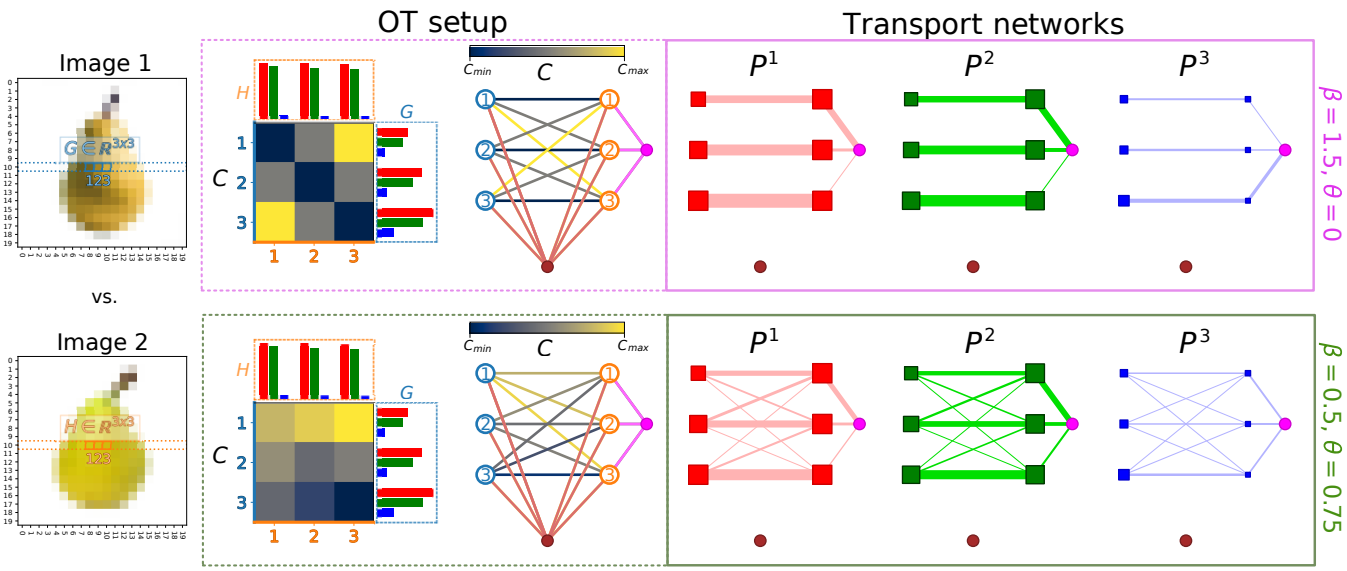

Figure S8. Effect of β on OT. In the leftmost portion of the panel we plot Image 1 and Image 2, used in the OT problem. From these we extract the (3×3)-dimensional tensors G and H. These are drawn together with a heatmap of the cost C, and with the correspondent transport network. Color scales of edges and of entries of C are identical. We also use the same numbering and color scheme for tensors' entries, indexes of C, and network nodes. Brown and magenta auxiliary nodes and edges are added after trimming, and after relaxing Kirchhoff's law. On the right side of the panel we plot the transport network again, but with edge thickness proportional to the Optimal Transport paths retrieved from Eqs. (3)-(4) (main text), and with colors correspondent to those of the commodities a. Node sizes are proportional to the values of g^a and h^a, for $a = 1, \ldots, 3$.

($\beta = 0.75$) naturally reflects the topology of the transport network and allows to achieve better classification performances.

Lastly, we remark that transport paths do not pass through any transshipment edge (colored in brown in Fig. S8) since τ is conveniently set to be sufficiently large. The auxiliary edges for Kirchhoff's law relaxation (colored in magenta) are instead traversed by transport paths since G and H are not normalized.

8 COMPUTATIONAL COST

8.1 Analytical discussion

Considerable effort has been spent to reduce the high complexity burden of OT problems. The $O(|V|^2/\varepsilon^3)$ baseline of Sinkhorn algorithm [11, 12, 10], where $|V|$ is the size of the histograms transported and ε the parameter enforcing entropic regularization, is constantly improved. Notable recent results are the class of stochastic optimization algorithms proposed in [13], that have been ameliorated using greedy alternatives [14] to achieve ε-approximation of the 1-Wasserstein distance between two probability distributions in $O(|V|^2/\varepsilon^2)$ arithmetic operations [15]. Recently, an Adaptive Primal-Dual Accelerated Gradient Descent (APDAMD) scheme with complexity $O(\min\{|V|^{9/4}/\varepsilon, |V|^2/\varepsilon^2\})$ for the same ε-perturbed problem has been presented in [16].

In principle, our multicommodity method has a computational complexity of order $O(M|V|^2)$ for complete transport graph topologies, i.e., when edges in the transport network K are assigned to

all pixels' pairs. Nonetheless, we achieve a substantial decrease in complexity by sparsifying the graph with the trimming procedure of [1, 2]. Similarly to [1], the final complexity of our algorithm is $O(M|V|)$. This improvement can be formally justified as follows, we start from a complete bipartite graph with $|E| = |V|^2/4$ (for simplicity $m = n = |V|/2$ is assumed). First, we trim expensive links, and reduce the number of edges of the transport network to $\langle K\rangle|V| + |V|$, where $\langle K\rangle$ is the average number of edges connected to a node that are not trimmed by τ, and the second term $|V|$ counts the number of inflowing and outflowing transshipment links. Second, we add $|V|/2$ links to the transport network to enforce Kirchhoff's law penalization, so that the final number of links amounts to $|E| = |V|(\langle K\rangle + 3/2)$, which is linear with respect to the number of nodes.

Additionally, it is shown [17] that for $\beta = 1$, the Optimal Transport paths of the unicommodity OT problem on sparse topologies can be recovered with z time steps as in Eq. (3) (main text), with $O(1) < z < O(|E|^{0.36})$. This bound has been found using a backward Euler scheme combined with the inexact Newton-Raphson method for the update of x, and solving Kirchhoff's law using an algebraic multigrid method.

8.2 Experimental runtimes benchmarking against Sinkhorn

We compare the runtime performances of the multicommodity dynamics of Eqs. (3)-(4) (main text), against the regularized Sinkhorn algorithm of [18, 19], implemented in POT: Python Optimal Transport [20], and for which we set the convergence threshold to $\widetilde{\varepsilon}_{\text{sink}} = 0.01$. Our implementation uses a forward Euler scheme for the discretization of Eq. (4) (main text), and a sparse direct linear solver (UMFPACK) for Eq. (3) (main text). Our code was run until convergence, achieved if $(J_\Gamma(n+1) - J_\Gamma(n))/\Delta t < \widetilde{\varepsilon}_{\text{dyn}}$, i.e. when the relative cost difference evaluated at two consecutive iteration is below $\widetilde{\varepsilon}_{\text{dyn}} = 1$. We set the discretization time step $\Delta t = 0.5$.

All codes are executed on 20 pairs of images, randomly sampled from the Jena Flowers 30 Dataset [7] and the Fruit Dataset [8]. We compare our multicommodity dynamics ($M = 3$) against Sinkhorn algorithm on colored images, and the unicommodity dynamics ($M = 1$) against Sinkhorn on grayscale images.

Results are shown in Fig. S9. Here, we plot elapsed times for the experiments on JF30 and on FD in the panels (A)-(C) and (D)-(F), respectively. Subplots from left to right represent runtimes for the algorithms executed on grayscale images [(A), (D)], on colored images [(B), (E)], and for Sinkhorn executed in both setups [(C), (F)].

Observing Fig. S9 [(A), (D)], we notice that runtimes for the multicommodity dynamics are larger than Sinkhorn's. Our algorithm converges faster if $\beta < 1$, i.e., when the multicommodity transport cost is convex. Setting $\beta > 1$ negatively affects convergence times. In general, for all values of β, increasing the trimming threshold τ, and thus the average number of edges in the transport networks, leads to slower convergence. Sinkhorn algorithm is not as dependent on $|E|$, e.g., in Fig. S9 (A), runtimes are approximately constant. Moreover, coherently to what expected [10], increasing the effect of the entropic barrier—enlarging ε—makes the algorithm faster. In Fig. S9 [(B), (E)] we observe a similar trend as in Fig. S9 [(A), (D)]. However, in this case Sinkorn algorithm with low regularization, $\varepsilon = 100$, has runtimes comparable to those of our method. Lastly, in Fig. S9 [(C), (F)], we explicitly plot runtimes for Sinkhorn on both colored and grayscale images, for different values of the regularization parameter ε. In general, the algorithm on colored images is slower, and increasing the trimming threshold leads to higher runtimes. Moreover, we observe again that larger value of ε makes the algorithms faster.

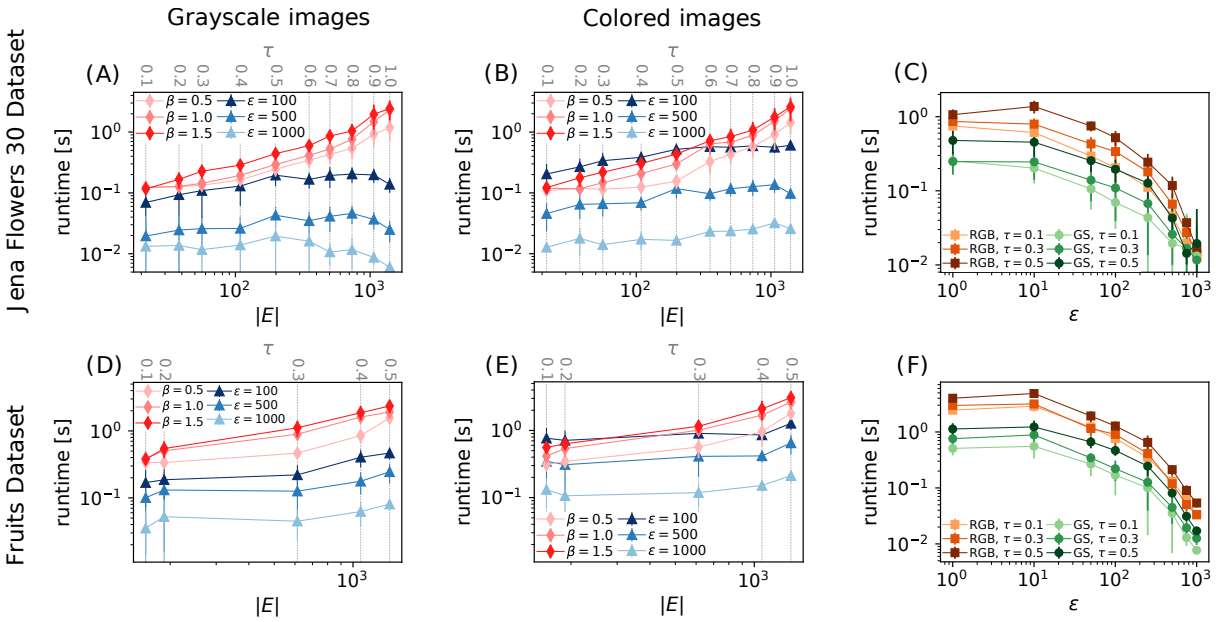

Figure S9. Runtimes of algorithms. Subplots (**A**)-(**C**) are experiments on JF30, subplots (**D**)-(**F**) are those on FD. In (A), (B), (C), and (D) we plot with red diamonds runtimes of our dynamics, with $M = 1$ in (A), (D) and $M = 3$ in (B), (E). Blue triangles are denote runtimes of Sinkhorn. Color shades correspond to different values of the regularization paramters. In (C) and (F) we show runtimes of Sinkhorn against ε, with orange and green markers used for colored and grayscale images, respectively. Color shades here denote different values of the trimming threshold τ. Errorbars are standard deviations obtained over 20 random image pairs.

REFERENCES

1 .Pele O, Werman M. A Linear Time Histogram Metric for Improved SIFT Matching. *Computer Vision – ECCV 2008* (Berlin, Heidelberg: Springer Berlin Heidelberg) (2008), 495–508. doi:10.1007/978-3-540-88690-7_37.

2 .Pele O, Werman M. Fast and robust Earth Mover's Distances. *2009 IEEE 12th International Conference on Computer Vision* (2009), 460–467. doi:10.1109/ICCV.2009.5459199.

3 .Shepard RN. Toward a Universal Law of Generalization for Psychological Science. *Science* **237** (1987) 1317–1323. doi:10.1126/science.3629243.

4 .Lonardi A, Facca E, Putti M, De Bacco C. Designing optimal networks for multicommodity transport problem. *Phys. Rev. Research* **3** (2021) 043010. doi:10.1103/PhysRevResearch.3.043010.

5 .Bonifaci V, Facca E, Folz F, Karrenbauer A, Kolev P, Mehlhorn K, et al. Physarum-inspired multi-commodity flow dynamics. *Theoretical Computer Science* (2022). doi:10.1016/j.tcs.2022.02.001.

6 .Bonifaci V, Mehlhorn K, Varma G. Physarum can compute shortest paths. *Journal of Theoretical Biology* **309** (2012) 121 – 133. doi:10.1016/j.jtbi.2012.06.017.

7 .[Dataset] Seeland M, Rzanny M, Alaqraa N, Wäldchen J, Mäder P. Jena Flowers 30 Dataset (2017). doi:10.7910/DVN/QDHYST.

8 .Macanhã PA, Eler DM, Garcia RE, Junior WEM. Handwritten feature descriptor methods applied to fruit classification. *Information Technology - New Generations* (Cham: Springer International Publishing) (2018), 699–705. doi:10.1007/978-3-319-54978-1_87.

9 .Poynton C. Frequently Asked Questions about Color. (1997).

10 .Cuturi M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems* (Curran Associates, Inc.) (2013), vol. 26, 2292–2300.

11 .Sinkhorn R. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics* **35** (1964) 876–879. doi:10.1214/aoms/1177703591.

12 .Knopp P, Sinkhorn R. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics* **21** (1967) 343 – 348. doi:pjm/1102992505.

13 .Genevay A, Cuturi M, Peyré G, Bach F. Stochastic optimization for large-scale optimal transport. Lee D, Sugiyama M, Luxburg U, Guyon I, Garnett R, editors, *Advances in Neural Information Processing Systems* (Curran Associates, Inc.) (2016), vol. 29.

14 .Altschuler J, Niles-Weed J, Rigollet P. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors, *Advances in Neural Information Processing Systems* (Curran Associates, Inc.) (2017), vol. 30.

15 .Lin T, Ho N, Jordan M. On Efficient Optimal Transport: An Analysis of Greedy and Accelerated Mirror Descent Algorithms. *Proceedings of the 36th International Conference on Machine Learning* (PMLR) (2019), *Proceedings of Machine Learning Research*, vol. 97, 3982–3991.

16 .Dvurechensky P, Gasnikov A, Kroshnin A. Computational Optimal Transport: Complexity by Accelerated Gradient Descent Is Better Than by Sinkhorn's Algorithm. *Proceedings of the 35th International Conference on Machine Learning* (PMLR) (2018), *Proceedings of Machine Learning Research*, vol. 80, 1367–1376.

17 .Facca E, Benzi M. Fast Iterative Solution of the Optimal Transport Problem on Graphs. *SIAM Journal on Scientific Computing* **43** (2021) A2295–A2319. doi:10.1137/20M137015X.

18 .Schmitzer B. Stabilized Sparse Scaling Algorithms for Entropy Regularized Transport Problems. *SIAM Journal on Scientific Computing* **41** (2019) A1443–A1481. doi:10.1137/16M1106018.

19 .Chizat L, Peyré G, Schmitzer B, Vialard FX. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation* **87** (2018) 2563–2609. doi:10.1090/mcom/3303.

20 .Flamary R, Courty N, Gramfort A, Alaya MZ, Boisbunon A, Chambon S, et al. POT: Python Optimal Transport. *Journal of Machine Learning Research* **22** (2021) 1–8.