Scalable Nearest Neighbor Search for Optimal Transport: Supplementary Material

Arturs Backurs [1 2] **Yihe Dong** [3] **Piotr Indyk** [4] **Ilya Razenshteyn** [5] **Tal Wagner** [4]

[1]Author names are ordered alphabetically. Code available at https://github.com/ilyaraz/ot_estimators.
[2]TTIC [3]Microsoft [4]MIT [5]Microsoft Research. Correspondence to: Tal Wagner <talw@mit.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

A. Proofs

A.1. Flowtree Computation

In this section we prove Lemma 1. We begin by specifying the greedy flow computation algorithm on the tree. Let h denote the height of the tree (for the quadtree the height is $h = O(\log(d\Phi))$. Suppose we are given a pair of distributions μ, ν, each supported on at most s leaves of the tree. For every node v in the tree, let $C_\mu(v)$ denote the set of points in $x \in X$ such that $\mu(x) > 0$ and the tree leaf that contains x is a descendant of v. Similarly define $C_\nu(v)$. Note that we only need to consider nodes for which either $C_\mu(v)$ or $C_\nu(v)$ is non-empty, and there are at most $2sh$ such nodes.

The algorithm starts with a zero flow f, and processes the nodes in a bottom-up order starting at the leaf. In each node, the unmatched demands collected from its children are matched arbitrarily, and the demands that cannot be matched are passed on to the parent. In mode detail, a node is processed as follows:

1. Collect from the children the list of unmatched μ-demands for the nodes in $C_\mu(v)$ and the list of unmatched ν-demands for the nodes in $C_\nu(v)$. Let $\{\mu_v(x) : x \in C_\mu(v)\}$ denote the unmatched ν-demands and let $\{\nu_v(x) : x \in C_\nu(v)\}$ denote the unmatched ν-demands.

2. While there is a pair $x \in C_\mu(v)$ and $x' \in C_\mu(v)$ with $\mu_v(x) > 0$ and $\nu_v(x') > 0$, let $\eta = \min\{\mu_v(x), \nu_v(x')\}$, and update (i) $f(x, x') \mathrel{+}= \eta$, (ii) $\mu_v(x) \mathrel{-}= \eta$, (iii) $\nu_v(x') \mathrel{-}= \eta$.

3. Now either μ_v or ν_v is all-zeros. If the other one is not all-zeros (i.e., there is either remaining unmatched μ-demand or remaining unmatched ν-demand), pass it on the the parent.

A leaf v contains a single point $x \in X$ with either $\mu(x) > 0$ or $\nu(x) > 0$; it simply passes it on to its parent without processing.

It is well known that the above algorithm computes an optimal flow on the tree (with respect to tree distance costs), see, e.g., (Kalantari & Kalantari, 1995). Let us now bound its running time. The processing time per node v in the above algorithm is $O(|C_\mu(v)| + |C_\nu(v)|)$. In every given level in the tree, if $v_1, \ldots, v_k$ are the nodes in that level, then $\{C_\mu(v_1), \ldots, C_\mu(v_k)\}$ is a partition of the support of μ, and $\{C_\nu(v_1), \ldots, C_\nu(v_k)\}$ is a partition of the support of ν. Therefore the total processing time per level is $O(s)$, and since there are h levels, the flow computation time is $O(sh)$. Then we need to compute the Flowtree output $\widetilde{W}_1(\mu, \nu)$. Observe that in the above algorithm, whenever we match demands between a pair x, x', we fully satisfy the unmatched demand of one of them. Therefore the output flow f puts non-zero flow between at most $2s$ pairs. For each such pair we need to compute the Euclidean distance in time $O(d)$, and the overall running time is $O(s(d + h))$.

A.2. Quadtree and Flowtree Analysis

Proof of Theorem 1. Let $x, y \in X$. Let $p_\ell(x, y)$ be the probability that x, y fall into the same cell (hypercube) in level ℓ of the quadtree. It satisfies,

$$1 - \frac{\|x - y\|_1}{2^\ell} \le p_\ell(x, y) \le \exp\left(-\frac{\|x - y\|_1}{2^\ell}\right). \tag{1}$$

To see this, recall that in level ℓ we impose a grid with side length 2^ℓ, shifted at random by an i.i.d. uniform shift in $[0, 2^\ell]$ in each coordinate. The probability that x, y are separated in coordinate i is $2^{-\ell}|x_i - y_i|$, and thus $p_\ell(x, y) = \prod_{i=1}^d (1 - 2^{-\ell}|x_i - y_i|)$. The lower bound in Equation (2) follows by a union bound, and the upper bound follows by applying the general estimate $1 - z \le \exp(-z)$ to each term in the product.

Let t be the tree metric induced on X by the quadtree. Note that for $t(x, y)$ to be at most $O(2^\ell)$, x, y must fall into the same hypercube in level ℓ. For any $\delta > 0$, we can round $\frac{\|x-y\|_1}{\log(1/\delta)}$ to its nearest power of 2 and obtain ℓ such that $2^\ell = \Theta(\frac{\|x-y\|_1}{\log(1/\delta)})$. It satisfies,

$$\Pr\left[t(x, y) < \frac{O(1)}{\log(1/\delta)} \|x - y\|_1\right] \le \delta.$$

By letting $\delta = \Omega(\min\{1/|X|, 1/(s^2 n)\})$, we can take union bound either over all pairwise distances in X (of which there are $\binom{|X|}{2}$), or over all distances between the support of the query ν and the union of supports of the dataset $\mu_1, \ldots, \mu_n$ (of which there are at most $s^2 n$, if every support has size at most s). Then, with probability say 0.995, all those distances are contracted by at most $O(\log(\min\{sn, |X|\}))$, i.e.,

$$t(x,y) \geq \frac{1}{O(\log(1/\delta))} \|x-y\|_1. \tag{2}$$

On the other hand,

$$\mathbb{E}[t(x,y)] = \sum_\ell 2^\ell \cdot (1 - p_\ell(x,y)) \leq \sum_\ell 2^\ell \cdot \frac{\|x-y\|_1}{2^\ell} \leq O(\log(d\Phi)) \cdot \|x-y\|_1.$$

Let μ^* be the true nearest neighbor of ν in $\mu_1, \dots, \mu_n$. Let $f^*_{\mu^*,\nu}$ be the optimal flow between them. Then by the above,

$$\mathbb{E}\left[\sum_{(x,y)\in X\times X} f^*_{\mu^*,\nu}(x,y)t(x,y)\right] \leq O(\log(d\Phi)) \sum_{(x,y)\in X\times X} f^*_{\mu^*,\nu}(x,y)\|x-y\|_1.$$

By Markov, with probability say 0.995,

$$\sum_{(x,y)\in X\times X} f^*_{\mu^*,\nu}(x,y) \cdot t(x,y) \leq O(\log(d\Phi)) \sum_{(x,y)\in X\times X} f^*_{\mu^*,\nu}(x,y) \cdot \|x-y\|_1. \tag{3}$$

Let μ' be the nearest neighbor of ν in the dataset according to the quadtree distance. Let $f^*_{\mu',\nu}$ be the optimal flow between them in the true underlying metric (ℓ_1 on X), and let $f_{\mu,\nu}$ be the optimal flow in the quadtree. Finally let W_t denote the Wasserstein-1 distance on the quadtree. Then,

$$\begin{aligned}
&W_1(\mu',\nu) \\
&= \sum_{(x,y)\in X\times X} f^*_{\mu',\nu}(x,y) \cdot \|x-y\|_1 \\
&\leq \sum_{(x,y)\in X\times X} f_{\mu',\nu} \cdot \|x-y\|_1 && f^*_{\mu^*,\nu} \text{ is optimal for } \|\cdot\|_1 \\
&\leq O(\log(\min\{sn,|X|\})) \sum_{(x,x')\in X\times X} f_{\mu',\nu} \cdot t(x,y) && \text{eq. (3)} \\
&= O(\log(\min\{sn,|X|\})) \cdot W_t(\mu',\nu) && \text{definition of } W_t \\
&\leq O(\log(\min\{sn,|X|\})) \cdot W_t(\mu^*,\nu) && \mu' \text{ is the nearest neighbor in } W_t \\
&= O(\log(\min\{sn,|X|\})) \sum_{(x,y)\in X\times X} f_{\mu^*,\nu} \cdot t(x,y) && \text{definition of } W_t \\
&\leq O(\log(\min\{sn,|X|\})) \sum_{(x,y)\in X\times X} f^*_{\mu^*,\nu} \cdot t(x,y) && f_{\mu^*,\nu} \text{ is optimal for } t(\cdot,\cdot) \\
&\leq O(\log(\min\{sn,|X|\}) \log(d\Phi)) \sum_{(x,y)\in X\times X} f^*_{\mu^*,\nu} \cdot \|x-y\|_1 && \text{eq. (4)} \\
&= O(\log(\min\{sn,|X|\}) \log(d\Phi)) \cdot W_1(\mu^*,\nu),
\end{aligned}$$

so μ' is a $O(\log(\min\{sn,|X|\}) \log(d\Phi))$-approximate nearest neighbor. □

Proof of Theorem 2. It suffices to prove the claim for $s = 1$ (i.e., the standard ℓ_1-distance). Let $d > 0$ be an even integer. Consider the d-dimensional hypercube. Our query point is the origin. The true nearest neighbor is e_1 (standard basis vector). The other data points are the hypercube nodes whose hamming weight is exactly $d/2$. The number of such points is $\Theta(2^d/\sqrt{d})$, and this is our n.

Consider imposing the grid with cell side 2 on the hypercube. The probability that 0 and 1 are uncut in a given axis is exactly $1/2$, and since the shifts in different axes are independent, the number of uncut axes is distributed as $Bin(d, 1/2)$.

Thus with probability $1/2$ there are at least $d/2$ uncut dimensions. If this happens, we have a data point hashed into the same grid cell as the origin (to get such data point, put 1 in any $d/2$ uncut dimensions and 0 in the rest), so its quadtree distance from the origin is 1. On the other hand, the distance of the origin to its true nearest neighbor e_1 is at least 1, since they will necessarily be separated in the next level (when the grid cells have side 1). Thus the quadtree cannot tell between the true nearest neighbor and the one at distance $d/2$, and we get the lower bound $c \geq d/2$. Since $n = \Theta(2^d/\sqrt{d})$, we have $d/2 = \Omega(\log n)$ as desired. □

Proof of Theorem 3. The proof is the same as for Theorem 1, except that in eq. (3), we take a union bound only over the s^2 distances between the supports of ν and μ^* (the query and its true nearest neighbor). Thus each distance between μ^* and ν is contracted by at most $O(\log s)$.

Let W_F denote the Flowtree distance estimate of W_1. Let μ' be the nearest neighbor of ν in the Flowtree distance. With the same notation in the proof of Theorem 1,

$$
\begin{aligned}
W_1(\mu',\nu) &= \sum_{(x,y)\in X\times X} f^*_{\mu',\nu}(x,y)\cdot\|x-y\|_1 \\
&\leq \sum_{(x,y)\times X\times X} f_{\mu',\nu}(x,y)\cdot\|x-y\|_1 && f^*_{\mu',\nu} \text{ is optimal for } \|\cdot\|_1 \\
&= W_F(\mu',\nu) && \text{Flowtree definition} \\
&\leq W_F(\mu^*,\nu) && \mu' \text{ is nearest in Flowtree distance} \\
&= \sum_{(x,y)\in X\times X} f_{\mu^*,\nu}(x,y)\cdot\|x-y\|_1 && \text{Flowtree definition} \\
&\leq O(\log s)\sum_{(x,y)\in X\times X} f_{\mu^*,\nu}(x,y)\cdot t(x,y) && \text{eq. (3)} \\
&\leq O(\log s)\sum_{(x,y)\in X\times X} f^*_{\mu^*,\nu}(x,y)\cdot t(x,y) && f_{\mu^*,\nu} \text{ is optimal for } t(\cdot,\cdot) \\
&\leq O(\log(d\Phi)\log s)\sum_{(x,y)\in X\times X} f^*_{\mu^*,\nu}(x,y)\cdot\|x-y\|_1 && \text{eq. (4)} \\
&= O(\log(d\Phi)\log s)\cdot W_1(\mu^*,\nu),
\end{aligned}
$$

as needed. Note that the difference from the proof of Theorem 1 is that we only needed the contraction bound (eq. (3)) for distances between μ^* and ν. □

Proof of Theorem 4. We set $\varepsilon = 1/\log s$. Let $t'(x,y)$ denote the quadtree distance where the weight corresponding to a cell v in level $\ell(v)$ is $2^{\ell(v)(1-\varepsilon)}$ instead of $2^{\ell(v)}$. Let $f_{\mu,\nu}$ be the optimal flow in the quadtree defined by weights t'.

Let $\delta = c/s^2$ where $c > 0$ is a sufficiently small constant. For a every x, y, let ℓ_{xy} be the largest integer such that

$$2^{\ell_{xy}} \leq \frac{\|x-y\|_1}{(\log(1/\delta))^{1/(1-\epsilon)}}.$$

The probability that x, y are separated (i.e., they are in different quadtree cells) in level ℓ_{xy} is

$$1 - p_{\ell_{xy}(x,y)} \geq 1 - \exp\left(-\frac{\|x-y\|_1}{2^{\ell_{xy}}}\right) \geq 1 - \frac{\delta}{1-\epsilon}.$$

By the setting of δ, we can take a union bound over all $x \in \mathrm{support}(\mu^*)$ and $y \in \mathrm{support}(\nu)$ and obtain that with say 0.99 probability, simultaneously, every pair x, y is separated at level ℓ_{xy}. We denote this event by $\mathcal{E}_{lower}$ and suppose it occurs. Then for every x, y we have

$$t'(x,y) \geq 2\cdot 2^{\ell_{xy}(1-\epsilon)} \geq 2\cdot\left(\frac{1}{2}\cdot\frac{\|x-y\|_1}{(\log(1/\delta))^{1/(1-\epsilon)}}\right)^{1-\epsilon} \geq \frac{\|x-y\|_1^{1-\epsilon}}{\log(1/\delta)} = \frac{\|x-y\|_1^{1-\epsilon}}{\Theta(\log s)}.$$

Next we upper-bound the expected tree distance $t'(x,y)$. (Note that we are not conditioning on $\mathcal{E}_{lower}$.) Observe that

$$t'(x,y) = 2\sum_{\ell=-\infty}^{\infty} 2^{\ell(1-\epsilon)} \cdot \mathbf{1}\{x,y \text{ are separated at level } \ell\}.$$

Let $L_{x,y}$ be the largest integer such that $2^{L_{xy}} \le \|x-y\|_1$. We break up $t'(x,y)$ into two terms,

$$t'_{lower}(x,y) = 2\sum_{\ell=-\infty}^{L_{xy}} 2^{\ell(1-\epsilon)} \cdot \mathbf{1}\{x,y \text{ are separated at level } \ell\},$$

and

$$t'_{upper}(x,y) = 2\sum_{\ell=L_{xy}+1}^{\infty} 2^{\ell(1-\epsilon)} \cdot \mathbf{1}\{x,y \text{ are separated at level } \ell\},$$

thus $t'(x,y) = t'_{lower}(x,y) + t'_{upper}(x,y)$. For $t'_{lower}(x,y)$ it is clear that deterministically,

$$t'_{lower}(x,y) \le 2\sum_{\ell=-\infty}^{L_{xy}} 2^{\ell(1-\epsilon)} = O\left(2^{L_{xy}(1-\epsilon)}\right) = O\left(\|x-y\|_1^{1-\epsilon}\right).$$

For $t'_{upper}(x,y)$, we have

$$\begin{aligned}
\mathbb{E}[t'_{upper}(x,y)] &= 2\sum_{\ell=L_{xy}+1}^{\infty} 2^{\ell(1-\epsilon)} p_\ell(x,y) \\
&\le 2\sum_{\ell=L_{xy}}^{\infty} 2^{\ell(1-\epsilon)} \cdot \frac{\|x-y\|_1}{2^\ell} \\
&= 2\|x-y\|_1 \sum_{\ell=L_{xy}}^{\infty} 2^{-\epsilon\ell} \\
&= 2\|x-y\|_1 \cdot \frac{2^{-L_{xy}\cdot\epsilon}}{1-2^{-\epsilon}} \\
&\le O(\log s) \cdot \|x-y\|_1^{1-\epsilon},
\end{aligned}$$

where in the final bound we have used that $2^{L_{xy}} = \Theta(\|x-y\|_1)$ and $1-2^{-\epsilon} = \Theta(\epsilon) = \Theta(\log s)$. Together,

$$\mathbb{E}[t'(x,y)] = \mathbb{E}[t'_{lower}(x,y) + t'_{upper}(x,y)] \le \Theta(\log s) \cdot \|x-y\|_1^{1-\epsilon}. \tag{4}$$

Now we are ready to show the $O(\log^2 s)$ upper bound on the approximation factor. Below we will use the fact that every weight $f_{\mu^*,\nu}(x,y)$ in the flow is of the form $i/(s's'')$ for some integer $0 \le i \le s's''$. This follows from the assumption that each element in the support of every measure is an integer multiple of $1/s'$ or of $1/s''$ for some $1 \le s', s'' \le s$.

$$\begin{aligned}
W_1(\mu',\nu) &= \sum_{(x,y)\in X\times X} f^*_{\mu',\nu}(x,y) \cdot \|x-y\|_1 && f^*_{\mu',\nu} \text{ is optimal for } \|\cdot\|_1 \\
&\le \sum_{(x,y)\times X\times X} f_{\mu',\nu}(x,y) \cdot \|x-y\|_1 \\
&= W_F(\mu',\nu) && \text{Flowtree definition} \\
&\le W_F(\mu^*,\nu) && \mu' \text{ is nearest to } \nu \text{ in Flowtree distance} \\
&= \sum_{(x,y)\in X\times X} f_{\mu^*,\nu}(x,y) \cdot \|x-y\|_1 && \text{Flowtree definition}
\end{aligned}$$

$$
\begin{aligned}
&\leq \left(\sum_{(x,y)\in X\times X} f_{\mu^*,\nu}^{1-\varepsilon}(x,y)\cdot \|x-y\|_1^{1-\varepsilon} \right)^{1/(1-\varepsilon)} && \text{subadditivity of } (\cdot)^{1-\varepsilon} \\
&\leq O(1) \left(\sum_{(x,y)\in X\times X} f_{\mu^*,\nu}(x,y)\cdot \|x-y\|_1^{1-\varepsilon} \right)^{1/(1-\varepsilon)} && f_{\mu^*,\nu}(x,y) \geq 1/(s's'') \geq 1/s^2 \text{ or } f_{\mu^*,\nu}(x,y)=0 \\
&\leq O(\log s) \left(\sum_{(x,y)\in X\times X} f_{\mu^*,\nu}(x,y)\cdot t'(x,y) \right)^{1/(1-\varepsilon)} && \text{eq. (5)} \\
&\leq O(\log s) \left(\sum_{(x,y)\in X\times X} f^*_{\mu^*,\nu}(x,y)\cdot t'(x,y) \right)^{1/(1-\varepsilon)} && f_{\mu^*,\nu} \text{ is optimal for } t'(\cdot,\cdot) \\
&\leq O(\log^2 s) \left(\sum_{(x,y)\in X\times X} f^*_{\mu^*,\nu}(x,y)\cdot \|x-y\|_1^{1-\varepsilon} \right)^{1/(1-\varepsilon)} && \text{eq. (5)} \\
&\leq O(\log^2 s) \sum_{(x,y)\in X\times X} f^*_{\mu^*,\nu}(x,y)\cdot \|x-y\|_1 && \text{concavity of } (\cdot)^{1-\varepsilon} \text{ and } \sum f^*_{\mu^*,\nu}(x,y)=1 \\
&\leq O(\log^2 s)\cdot W_1(\mu^*,\nu),
\end{aligned}
$$

as needed. □

Proof of Theorem 5. **Quadtree.** For every $k = 1, \dots, s$, let H_k be the smallest hypercube in the quadtree that contains both x_k and y_k. (Note that H_k is a random variable, determined by the initial random shift in the Quadtree construction.) In order for Quadtree to correctly identify μ_i as the nearest neighbor of ν, every H_k must not contain any additional points from X. Otherwise, if say H_1 contains a point $x' \neq x_1$, the W_1 distance on the quadtree from ν to μ_i is equal to its distance to the uniform distribution over $\{x', x_2, \dots, x_s\}$. Since the points in X are chosen uniformly i.i.d. over $\mathcal{S}^{d-1}$, the probability of the above event, and thus the success probability of Quadtree, is upper bounded by $\mathbb{E}[(1-V)^{N-s}]$, where $V = \text{volume}(\cup_{k=1}^s H_k \cap \mathcal{S}^{d-1})$. This V is a random variable whose distribution depends only on d, s, ϵ, and is independent of N. Thus the success probability decays exponentially with N.

Flowtree. On the other hand, suppose that each H_k contains no other points from $\{x_1, \dots, x_s\}$ other than x_k (but is allowed to contain any other points from X). This event guarantees that the optimal flow on the tree between μ_i and ν is the planted perfect matching, i.e., the true optimal flow, and thus the estimated Flowtree distance between them *equals* $W_1(\mu_i, \nu)$. This guarantees that Flowtree recovers the planted nearest neighbor, and this event depends only on d, s, ϵ, and is independent of N. □

B. Additional Experiments

B.1. Additional Sinkhorn and ACT Experiments

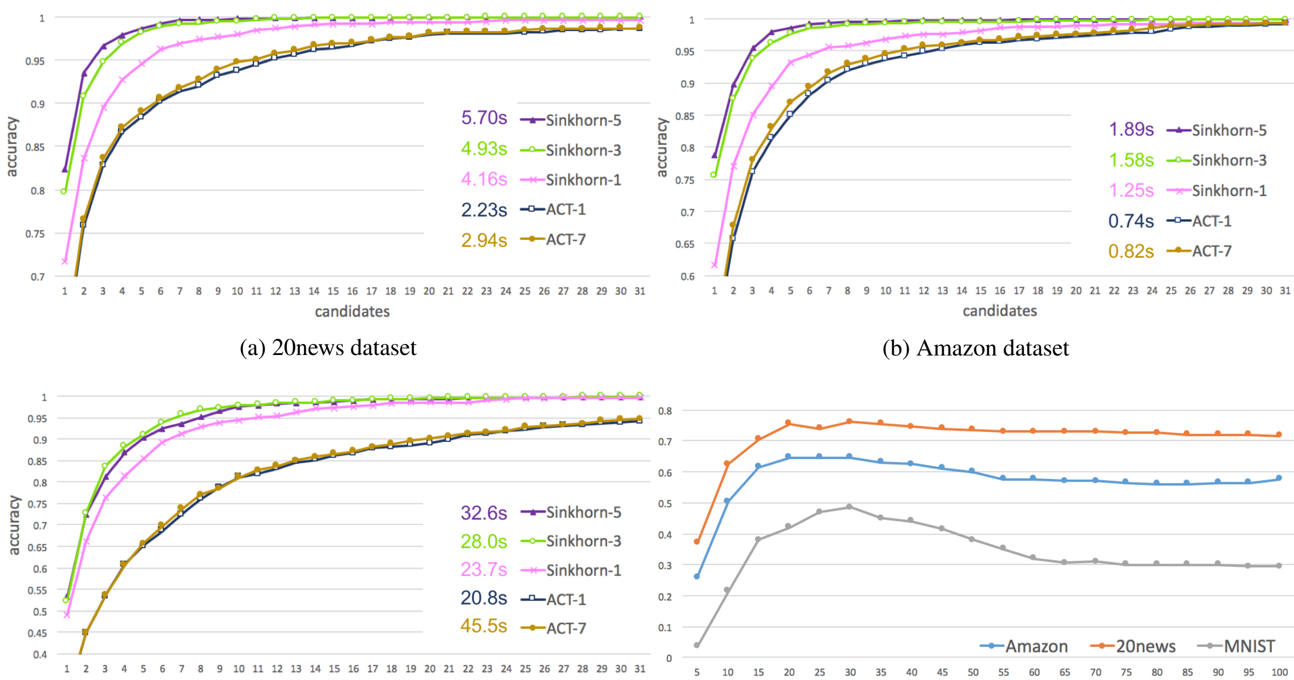
Number of iterations. Both ACT and Sinkhorn are iterative algorithms, and the number of iterations is a parameter to set. Our main experiments use ACT with 1 iteration and Sinkhorn with 1 or 3 iterations. The next experiments motivate these choices. Figures 7(a)–(c) depict the accuracy and running time of ACT-1, ACT-7, Sinkhorn-1, Sinkhorn-3 and Sinkhorn-5 on each of our datasets.[1] It can be seen that for both algorithms, increasing the number of iterations beyond the settings used in Section 4 yields comparable accuracy with a slower running time. Therefore in Section 4 we restrict our evaluation to ACT-1, Sinkhorn-1 and Sinkhorn-3. We also remark that in the pipeline experiments, we have evaluated Sinkhorn with up to 9 iterations. In those experiments too, the best results are achieved with either 1 or 3 iterations

Sinkhorn regularization parameter. Sinkhorn has a regularization parameter λ that needs to be tuned per dataset. We set $\lambda = \eta \cdot M$, where M is the maximum value in the cost matrix (of the currently evaluated pair of distributions), and

[1] ACT-1 and ACT-7 are the settings reported in (Atasu & Mittelholzer, 2019).

tune η. In all of our three datasets the optimal setting is $\eta = 30$, which is the setting we use in Section 4. As an example, Figure 7(d) depicts the 1-NN accuracy (y-axis) of Sinkhorn-1 per η (x-axis).

Figure 1: Additional Sinkhorn and ACT experiments



(a) 20news dataset

(b) Amazon dataset

(c) MNIST dataset

(d) 1-NN accuracy of Sinkhorn-1 with varying regularization

B.2. Additional Pipeline Results

The next tables summarize the running times and parameters settings of all pipelines considered in our experiments (whereas the main text focuses on pipelines that start with Quadtree, since it is superior as a first step to Mean and Overlap). The listed parameters are the number of output candidates of each step in the pipeline.

In the baseline pipelines, parameters are tuned to achieve optimal performance (i.e., minimize the running time while attaining the recall goal on at least 90% of the queries). The details of the tuning procedure is as follows. For all pipelines we use the same random subset of 300 queries for tuning. Suppose the pipeline has ℓ algorithms. For $i = 1, \ldots, \ell$, let c_i the output number of candidates of the ith algorithm in the pipeline. Note that c_ℓ always equals either 1 or 5, according to the recall goal of the pipeline, so we need to set $c_1, \ldots, c_{\ell-1}$. Let p_1 be the recall@1 accuracy of the first algorithm in the pipeline. Namely, p_1 is the fraction of queries such that the top-ranked c_1 candidates by the first algorithm contain the true nearest neighbor. We calculate 10 possible values of c_1, corresponding to $p_1 \in \{0.9, 0.91, \ldots, 0.99\}$. We optimize the pipeline by a full grid search over those values of c_1 and all possible values of $c_2, \ldots, c_{\ell-1}$.

When introducing Flowtree into a pipeline as an intermediate method, we do not re-optimize the parameters, but rather set its output number of candidates to the maximum between 10 and twice the output number of candidates of the subsequent algorithm in the pipeline. Re-optimizing the parameters could possibly improve results.

Pipeline methods	Candidates	Time
Mean, Sinkhorn-1, Exact	1476, 11, 1	0.543
Mean, Sinkhorn-3, Exact	1476, 5, 1	0.598
Mean, R-WMD, Exact	1850, 28, 1	0.428
Mean, ACT-1, Exact	1677, 14, 1	0.420
Overlap, Sinkhorn-1, Exact	391, 6, 1	0.610
Overlap, Sinkhorn-3, Exact	391, 5, 1	0.691
Overlap, R-WMD, Exact	576, 14, 1	0.367
Overlap, ACT-1, Exact	434, 10, 1	0.429
Quadtree, Sinkhorn-1, Exact	295, 5, 1	0.250
Quadtree, Sinkhorn-3, Exact	227, 3, 1	0.248
Quadtree, R-WMD, Exact	424, 12, 1	0.221
Quadtree, ACT-1, Exact	424, 8, 1	0.236

Table 1: Recall@1, no Flowtree.

Pipeline methods	Candidates	Time
Mean, **Flowtree**, Sinkhorn-1, Exact	1850, 10, 5, 1	0.089
Mean, **Flowtree**, Sinkhorn-3, Exact	1677, 10, 4, 1	0.077
Mean, **Flowtree**, R-WMD, Exact	2128, 48, 24, 1	0.242
Mean, **Flowtree**, ACT-1, Exact	2128, 20, 10, 1	0.138
Overlap, **Flowtree**, Sinkhorn-1, Exact	489, 10, 5, 1	0.087
Overlap, **Flowtree**, Sinkhorn-3, Exact	576, 10, 3, 1	0.076
Overlap, **Flowtree**, R-WMD, Exact	576, 28, 14, 1	0.173
Overlap, **Flowtree**, ACT-1, Exact	576, 16, 8, 1	0.119
Quadtree, **Flowtree**, Sinkhorn-1, Exact	424, 10, 5, 1	0.074
Quadtree, **Flowtree**, Sinkhorn-3, Exact	424, 10, 3, 1	0.059
Quadtree, **Flowtree**, R-WMD, Exact	424, 22, 11, 1	0.129
Quadtree, **Flowtree**, ACT-1, Exact	424, 16, 8, 1	0.104
Mean, **Flowtree**, Exact	1850, 9, 1	0.105
Overlap, **Flowtree**, Exact	489, 9, 1	0.100
Quadtree, **Flowtree**, Exact	424, 9, 1	0.092

Table 2: Recall@1, with Flowtree.

Pipeline methods	Candidates	Time
Mean, Sinkhorn-1	1476, 5	0.464
Mean, Sinkhorn-3	1476, 5	0.549
Mean, R-WMD, Exact	1850, 28, 5	0.426
Mean, ACT-1, Exact	1677, 14, 5	0.423
Overlap, Sinkhorn-1	391, 5	0.560
Overlap, Sinkhorn-3	391, 5	0.650
Overlap, R-WMD, Exact	576, 14, 5	0.368
Overlap, ACT-1, Exact	434, 10, 5	0.428
Quadtree, Sinkhorn-1	295, 5	0.222
Quadtree, Sinkhorn-3	227, 5	0.200
Quadtree, R-WMD, Exact	424, 11, 5	0.216
Quadtree, ACT-1, Exact	424, 7, 5	0.222

Table 3: Recall@5, no Flowtree.

Pipeline methods	Candidates	Time
Mean, **Flowtree**, Sinkhorn-1	1850, 10, 5	0.046
Mean, **Flowtree**, Sinkhorn-3	1476, 10, 5	0.043
Mean, **Flowtree**, R-WMD, Exact	2128, 48, 24, 5	0.237
Mean, **Flowtree**, ACT-1	2128, 10, 5	0.048
Overlap, **Flowtree**, Sinkhorn-1	391, 10, 5	0.042
Overlap, **Flowtree**, Sinkhorn-3	391, 10, 5	0.044
Overlap, **Flowtree**, R-WMD, Exact	576, 28, 14, 5	0.173
Overlap, **Flowtree**, ACT-1	576, 10, 5	0.046
Quadtree, **Flowtree**, Sinkhorn-1	424, 10, 5	0.033
Quadtree, **Flowtree**, Sinkhorn-3	424, 10, 5	0.034
Quadtree, **Flowtree**, ACT-1	424, 10, 5	0.029
Mean, **Flowtree**	2128, 5	0.043
Overlap, **Flowtree**	576, 5	0.039
Quadtree, **Flowtree**	645, 5	0.027
Quadtree, **Flowtree**, R-WMD, Exact	424, 22, 11, 5	0.131
Quadtree, **Flowtree**, ACT-1, Exact	424, 16, 8, 5	0.103

Table 4: Recall@5, with Flowtree.