The Power of Bounds: Answering Approximate Earth Mover's Distance with Parametric Bounds

Tsz Nam Chan, Man Lung Yiu, and Leong Hou U, *Member, IEEE*

Abstract—The *Earth Mover's Distance* (EMD) is a robust similarity measure between two histograms (e.g., probability distributions). It has been extensively used in a wide range of applications, e.g., multimedia, data mining, computer vision, etc. As EMD is a computationally intensive operation, many efficient lower and upper bound functions of EMD have been developed. However, they provide no guarantee on the error. In this work, we study how to compute approximate EMD value with bounded error. First, we develop a parametric dual bound function for EMD, in order to offer sufficient trade-off points for optimization. After that, we propose an approximation framework that leverages on lower and upper bound functions to compute approximate EMD with error guarantee. Then, we present three solutions to solve our problem. Experimental results on real data demonstrate the efficiency and the effectiveness of our proposed solutions.

Index Terms—Earth mover's distance, parametric bounds, approximation framework

1 Introduction

The *Earth Mover's Distance* (EMD) is a similarity measure between two histograms (e.g., probability distributions). It is more robust than traditional similarity measures like the Euclidean distance. EMD has been extensively used in multimedia databases [6], [7], [17], [20], [34], [39], [42], [44], data mining [5], computer vision [30], [36], [40], artificial intelligence [28], machine learning [10] and text retrieval [25]. Nevertheless, EMD is a computational intensive operation. Even with the fastest known algorithm [29], it requires $O(d^3 \log d)$ time to compute the exact EMD value, where d is the dimensionality (i.e., number of histogram bins). Furthermore, the need for rapid solutions is motivated by the fact that many applications require EMD computations on a massive amount of objects, which are quoted as follows:

- *"In real applications, datasets may contain hundreds of thousands or even millions of objects. An EMD similarity join on them may take weeks to months to complete on a single machine."* [17]
- *"Typically, the EMD between two histograms is modeled and solved as a linear optimization problem, the min-cost flow problem, which requires super-cubic time. The high computational cost of EMD restricts its applicability to datasets of low-scale."* [39]

- *T.N. Chan and M.L. Yiu are with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. E-mail: {cstnchan, csmlyiu}@comp.polyu.edu.hk.*
- *L.H. U is with the State Key Laboratory of Internet of Things for Smart City, Department of Computer and Information Science, University of Macau, Taipa, Macau, China. E-mail: ryanlhu@um.edu.mo.*

Manuscript received 9 Sept. 2018; revised 31 May 2019; accepted 16 July 2019. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Tsz Nam Chan.)
Recommended for acceptance by K. Yi.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2019.2931969

In this paper, we focus on computing approximate EMD yet allowing user to control the error. Specifically, given an error parameter ϵ, our problem is to find an approximate EMD value R such that R is within $1 \pm \epsilon$ times the exact EMD value. We are not aware of efficient algorithms that satisfy the above error requirement. The database community has derived several lower and upper bound functions of EMD [6], [20], [34], [39], [42]. However, these bound functions provide no guarantee on the error of the bound.

Motivated by this, we wonder *whether existing lower and upper bound functions can be exploited to solve our problem efficiently*. Intuitively, if we can obtain a lower bound ℓ and an upper bound u of the exact EMD value such that they are sufficiently close (e.g., $u/\ell \leq 1+\epsilon$), then we get an approximate EMD value with error guarantee. The next question is how to select appropriate lower and upper bound functions with respect to ϵ. Consider all possible pairs of $\langle LB_i, UB_j \rangle$ where LB_i is a lower bound function, and UB_j is an upper bound function. Ideally, if we can accurately estimate the response time and the error for each pair $\langle LB_i, UB_j \rangle$, as shown in Fig. 1, then the optimal solution is to choose the cheapest pair (i.e., $\langle LB_1, UB_4 \rangle$) whose error is below ϵ. The challenge is how to estimate quickly the response time and the error, while the estimates are reasonably accurate. This issue is complicated by the fact that, even within the same dataset, the same pair $\langle LB_i, UB_j \rangle$ of bound functions may yield different response time and error for different pairs of histograms.

Another issue is that, the limited number of bound functions prevents us to conduct fine-grained optimization. We need a wide spectrum of bound functions for EMD to provide sufficient trade-off points for optimization. To address this issue, we propose the concept of *parametric dual bound function*, which produces both lower and upper bounds simultaneously via shared computation, while its running time and tightness can be controlled via a parameter. In Fig. 1, we indicate a parametric dual bound function by a dotted line in blue. By choosing its parameter value

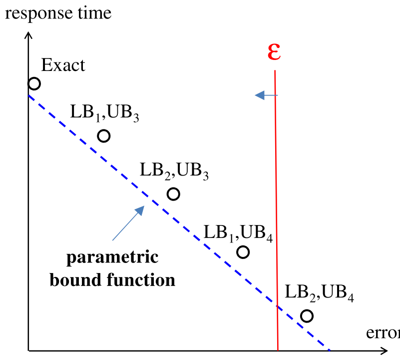


Fig. 1. Illustration of bound functions.

carefully, it is possible to obtain a better choice than the pair $\langle LB_1, UB_4\rangle$. Since it is common to have skewed data in real applications [16], [17], we will exploit the characteristics of skewed data to design a parametric dual bound function. As a remark, Wichterich et al. [42] have devised a parametric lower bound function, but not any parametric upper bound function. In contrast, we utilize shared computation to compute both lower and upper bounds simultaneously.

We attempt to tackle our problem in two directions. First, we propose an *adaptive* approach, which does not rely on any training. It gradually invokes tighter bound functions until satisfying the error requirement. Then, we develop an enhancement, called *lightweight adaptive* approach, by reducing the number of calls to bound functions. Finally, we apply training to collect statistics, and exploit them to boost the performance of our solution.

In our experimental study, we will evaluate both the efficiency and the effectiveness of our proposed methods. We will conduct case study on the representative application (i.e., kNN image retrieval) and demonstrate that approximate EMD values yield reasonably accurate results. Our methods achieve an order of magnitude speedup over the fastest exact computation method.

Our technical contributions are summarized as follows.

- We develop a parametric dual bound function for EMD, in order to offer sufficient trade-off points for optimization.
- We propose an approximation framework that leverages lower and upper bound functions to compute approximate EMD with error guarantee.
- We present three solutions for our problem via our approximation framework.

We first discuss the related work in Section 2. In Section 3, we define our problem formally and briefly review existing bound functions for EMD in the literature. In Section 4, we present our parametric lower and upper bound functions. We propose our approximation framework with three solutions in Section 5. In Section 6, we present experimental results on real datasets and then conclude in Section 7.

2 Related Work

The *Earth Mover's Distance* was first introduced in [32] as a similarity metric in image databases. Comparing to other bin-by-bin distances (e.g., Euclidean distance), the cross-bin calculation makes EMD better match the human perception of differences. EMD can be regarded as a special case of the minimum cost flow problem and many algorithms have been proposed in literature [2], e.g., capacity scaling algorithm, cost scaling algorithm, transportation simplex, and network simplex. However, their worst case time complexity remains super cubic to the number of bins, which limits the applicability of EMD.

In order to employ EMD as a similarity metric in large datasets, the database community attempted to use a filter-and-refinement framework to reduce the number of exact EMD computations. The key factor of the filter-and-refinement framework is to provide a tight lower/upper bound estimation such that more EMD computations can be pruned at the filtering stage. Thereby, there are plenty of EMD bounding techniques [6], [26], [34], [39], [42] being proposed in the database community. In this work, we design a new approximate framework by reutilizing these bounds, which not only provides high quality approximate result (due to the tightness of these bounds) but also reduces the implementation difficulty.

In the theoretical computer science community, there are quite a few of studies [1], [4], [19], [24] in calculating approximate EMD. However, they either focus on a planar graph setting (i.e., calculating EMD on two planar point-sets) [19] or lack of flow concept (i.e., the approximate ratio is analyzed based on a uni-flow model) [1], [4], [24]. On the other hand, EMD is also the special case of the minimum cost flow problem in which their approximation methods can be also applied for EMD. Some recent studies from [8], [31] can be used to compute approximate EMD within the δ-additive error, i.e., the approximate EMD value differs from the exact EMD value by at most δ. The time complexity of these algorithms normally depends on both the dimensionality, δ and the maximum value of the cost matrix. However, it is hard to set the parameter δ, since setting the reasonable δ (not too large or small) depends on the exact EMD value, which is not known in advance. In contrast, we propose to use ϵ-multiplicative error (i.e., relative error); it is easier to choose the parameter ϵ as it does not require knowing the exact EMD value. Recently, Sherman [35] proposes generalized preconditioning method to transform minimum cost flow problem to minimum $\ell 1$ norm problem, which can be efficiently solved by combining existing numerical solvers [9], [27]. This approach can be used to evaluate approximate EMD value within the relative error ϵ in $O(d^{2(1+o(1))}\epsilon^{-2})$ time (which is near $O(d^2\epsilon^{-2})$ once d is very large).

Approximate EMD has also been studied in the computer vision [30], [36], database [20] and machine learning [3] communities. Pele et al. [30] remove some records from the cost matrix when their values are larger than a pre-defined threshold. The EMD computation time is correlated to the sparsity of the cost matrix so that the threshold plays a role in controlling the quality and the efficiency. Jang et al. [20] store a set of hilbert curves and assign the distance between two images based on these curves. However, the approximate quality is highly relevant to the hilbert curve selection and there is no theoretical guarantee. Shirdhonkar et al. [36] utilize the wavelet theory in their approximation algorithm, but do not offer any error guarantee. Altschuler et al. [3] develop Sinkhorn projection-based iterative algorithms for evaluating approximate EMD. Like [8], [31], they provide

TABLE 1
Symbols

Symbols	Description
$emd_c(\mathbf{q},\mathbf{p})$	Earth Mover's Distance between vectors $\mathbf{q}$ and $\mathbf{p}$
$\mathbb{E}_{\mathbf{q},\mathbf{p}}(R)$	Relative error between R and $emd_c(\mathbf{q},\mathbf{p})$
LB, UB	Lower and upper bound functions
ℓ, u	Abbreviation of $LB(\mathbf{q},\mathbf{p})$ and $UB(\mathbf{q},\mathbf{p})$
$E_{max}(\ell, u)$	Validation function $E_{max}(\ell,u)=\frac{u-\ell}{u+\ell}$
Γ	Historical workload
$\mathbb{T}(LB(\mathbf{q},\mathbf{p}), UB(\mathbf{q},\mathbf{p}))$	Running time of LB and UB on $(\mathbf{q},\mathbf{p})$ pair
$\mathbb{T}(Alg(\mathbf{q},\mathbf{p}))$	Running time of Alg (e.g., ADA-L ...) on $(\mathbf{q},\mathbf{p})$ pair

the δ-additive error guarantee, which is hard to set in practice, but not relative error guarantee.

3 Preliminaries

3.1 Problem Definition

The *Earth Mover Distance* [33] can be used to measure the dissimilarity between two histograms (e.g., probability distributions). We represent a histogram by $\mathbf{p}=[p_1,p_2,\ldots,p_d]$, where d is the dimensionality (i.e., number of bins). Following the previous studies [33], [34], [39], we assume that each histogram $\mathbf{p}$ is normalized, i.e., $\sum_{j=1}^{d} p_j = 1$. Given two histograms $\mathbf{q}$ and $\mathbf{p}$, the EMD between them is defined as the *minimum-cost flow* on a bipartite flow network between $\mathbf{q}$ and $\mathbf{p}$. We denote a cost matrix by c and a flow matrix by f, where $c_{i,j}$ models the cost of moving flow from q_i to p_j, and $f_{i,j}$ represents the amount of flow to move from q_i to p_j. Formally, we define $emd_c(\mathbf{q},\mathbf{p})$ as the following linear programming problem.

$$\begin{aligned} emd_c(\mathbf{q},\mathbf{p}) = \quad & \underset{f}{\text{minimize}} \sum_{i=1}^{d}\sum_{j=1}^{d} c_{i,j} f_{i,j} \\ \text{such that} \quad & \forall i,j \in [1..d] : f_{i,j} \geq 0 \\ & \forall i \in [1..d] : \sum_{j=1}^{d} f_{i,j} = q_i \\ & \forall j \in [1..d] : \sum_{i=1}^{d} f_{i,j} = p_j. \end{aligned}$$

According to Ref. [33], EMD satisfies the triangle inequality provided that the cost matrix c is a metric (i.e., non-negativity, symmetry and triangle inequality for all $c_{i,j}$).

Lemma 1 (Proved in Ref. [33]). *For any histograms $\mathbf{q},\mathbf{p},\mathbf{r}$ with the same dimensionality, we have:*

$$emd_c(\mathbf{q},\mathbf{p}) \leq emd_c(\mathbf{q},\mathbf{r}) + emd_c(\mathbf{r},\mathbf{p}).$$

EMD is computationally expensive. Even with the fastest known algorithm [29], it is still expensive to compute the exact EMD value, which takes $O(d^3 \log d)$ time. Instead, we propose to compute an approximate EMD value R with bounded error. We formulate our problem below; it guarantees that R is within $1 \pm \epsilon$ times the exact EMD value. Our objective is to develop efficient algorithms for this problem.

TABLE 2
Summary of Lower and Upper Bound Functions for EMD

Name	Type	Time Complexity	Reference	Parametric
LB_{IM}	lower	$O(d^2)$	[6]	no
LB_{Proj}	lower	$O(d)$	[11], [34]	no
LB_{Red,d_r}	lower	$O(d^2 + {d_r}^3 \log d_r)$	[42]	yes
UB_G	upper	$O(d^2)$	[39]	no
UB_H	upper	$O(d)$	[20], [21]	no

Problem 1 (Error-Bounded EMD). *Given an error threshold ϵ, this problem returns a value R such that $\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) \leq \epsilon$, where the relative error of R is defined as*

$$\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) = \frac{|R - emd_c(\mathbf{q},\mathbf{p})|}{emd_c(\mathbf{q},\mathbf{p})}. \tag{1}$$

Table 1 summarizes the frequently-used symbols in this paper.

3.2 Existing Bound Functions

We introduce existing lower bound and upper bound functions for EMD in the literature, which will be used in subsequent sections. These bound functions must satisfy the following properties (cf. Definition 1).

Definition 1. *$LB(\mathbf{q},\mathbf{p})$ is called a lower bound function if $emd_c(\mathbf{q},\mathbf{p}) \geq LB(\mathbf{q},\mathbf{p})$ for any $\mathbf{q},\mathbf{p}$. $UB(\mathbf{q},\mathbf{p})$ is called an upper bound function if $emd_c(\mathbf{q},\mathbf{p}) \leq UB(\mathbf{q},\mathbf{p})$ for any $\mathbf{q},\mathbf{p}$.*

We summarize several representative lower and upper bound functions in the literature in Table 2. For each bound function, we show its name (in subscript), its time complexity, and its reference(s). We refer the interested readers to the references.

Most of the bound functions yield time complexities in terms of the histogram dimensionality d. Wichterich et al. [42] propose a *parametric* lower bound function LB_{Red,d_r}, which accepts an additional parameter d_r (i.e., reduced dimensionality) to control its running time and tightness.

4 Parametric Dual Bounding

Although there exists a parametric lower bound function [42], we are not aware of any parametric upper bound function in the literature. Instead of providing separate functions for lower bound and upper bound, we propose another *parametric dual bound* function, which utilizes shared computation to compute both lower and upper bounds simultaneously, and offers control on running time and tightness via a parameter. Moreover, our parametric bound functions take advantage of skewed property of data, we provide the case study in Section 4.4 to demonstrate our parametric lower bound is generally superior than [42] for small error in this type of datasets.

4.1 Exact EMD on Sparse Histograms

Recall from Section 3 that the computation of $emd_c(\mathbf{q},\mathbf{p})$ is equivalent to the minimum-cost flow problem on a bipartite

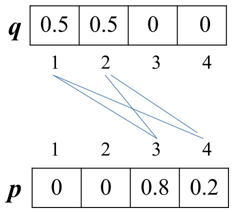


Fig. 2. Bipartite graph for sparse EMD computation.

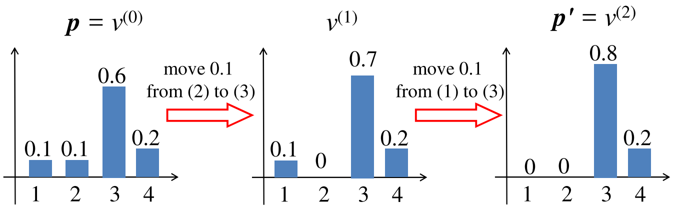


Fig. 3. Example for skew transform.

flow network. This flow network contains d^2 edges because there is an edge between each q_i and each p_j.

It turns out that, when both $\mathbf{q}$ and $\mathbf{p}$ are sparse histograms (i.e., having 0 in many bins), it is possible to shrink the flow network without affecting the exact EMD value. Consider the example in Fig. 2 where the dimensionality is $d = 4$. Since each flow $f_{i,j}$ (from q_i to p_j) must be non-negative, any bin with zero value must have zero flow (to and from other bins). Therefore, we can safely remove the edge for $f_{i,j}$ if either $q_i = 0$ or $p_j = 0$. For example, in Fig. 2, it suffices to keep only $2 \times 2 = 4$ edges in the flow network.

Formally, we introduce the notation $\Phi(\mathbf{p})$ to measure the denseness of the histogram.

Definition 2. *Let $\Phi(\mathbf{p})$ be the number of non-zero bins in histogram $\mathbf{p}$, i.e., $\Phi(\mathbf{p}) = \texttt{COUNT}\{j : p_j \neq 0\}$.*

With this idea, we can compute the exact value of $emd_c(\mathbf{q},\mathbf{p})$ in $O(d_s^3 \log d_s)$ time, where $d_s = \max(\Phi(\mathbf{q}), \Phi(\mathbf{p}))$.

4.2 Skew-Transform Operation

Based on the idea of efficient EMD evaluation on sparse histograms, we propose the *Skew-Transform* operation which will be used for our skew-based bound functions. This operation takes a histogram $\mathbf{p}$ and an integer λ as input, and returns a histogram $\mathbf{p}'$ that contains exactly λ non-zero bins (i.e., $\Phi(\mathbf{p}') = \lambda$). We illustrate an example of this operation in Fig. 3, with the input $\mathbf{p} = [0.1, 0.1, 0.6, 0.2]$ and $\lambda = 2$. After moving values in bin 1 and bin 2 to bin 3, we obtain the histogram $\mathbf{p}' = [0, 0, 0.8, 0.2]$, which contains exactly two non-zero bins. As we will explain later, an upper bound $ub_{move}(\mathbf{p},\mathbf{p}')$ can be derived efficiently from the sequence of movements. In this case, we have: $ub_{move}(\mathbf{p},\mathbf{p}') = 0.1 \cdot c_{2,3} + 0.1 \cdot c_{1,3}$.

We adopt a greedy method to implement the skew-transform operation. First, we select a source bin (say, s) with the smallest non-zero value. Then, we select a target bin (say, t) such that it has non-zero value and the smallest movement cost $c_{s,t}$. We repeat the above procedure until the result histogram $\mathbf{p}'$ contains exactly λ non-zero bins. The pseudocode of this method is described in Algorithm 1.

The value ub_{move} computed by Algorithm 1 is indeed an upper bound of $emd_c(\mathbf{p},\mathbf{p}')$. We prove this in the following lemma.

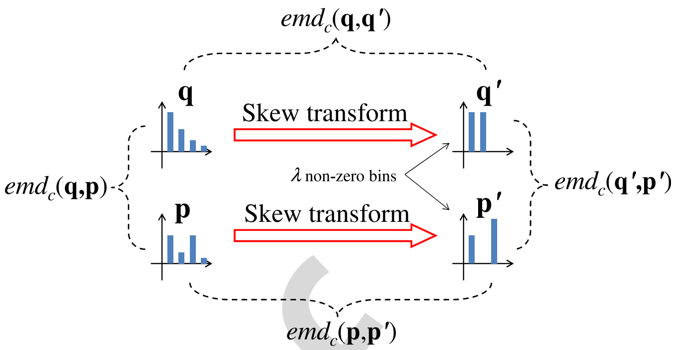


Fig. 4. Skew-based lower and upper bounds.

Algorithm 1. Skew Transform Operation

procedure SKEW-TRANSFORM (histogram $\mathbf{p}$, cost matrix c, integer λ)
  $\mathbf{p}' \leftarrow \mathbf{p}$
  $ub_{move} \leftarrow 0$
  while $\Phi(\mathbf{p}') > \lambda$ **do**
    $s \leftarrow \arg\min\{i : p'_i \neq 0\}$
    $t \leftarrow \arg\min\{j : c_{s,j}, p'_j \neq 0, j \neq s\}$
    $\delta \leftarrow p'_s$
    $ub_{move} \leftarrow ub_{move} + c_{s,t}\delta$
    $p'_t \leftarrow p'_t + \delta;\ p'_s \leftarrow 0$
  return $(\mathbf{p}', ub_{move})$

Lemma 2. *When Algorithm 1 terminates, it holds that: $ub_{move} \geq emd_c(\mathbf{p},\mathbf{p}')$.*

Proof. In each iteration of the Algorithm 1, the movement of value δ from bin s to bin t is feasible; it preserves the summation terms $\sum_{j=1..d} f_{ij} = p_i, \forall i$, $\sum_{i=1..d} f_{ij} = p'_j, \forall j$ and ensures that each movement incurs non-negative flow from bin s to t. Therefore, $f_{ij} \geq 0, \forall i,j$. This means that the total movement cost is at least the minimum possible cost $emd_c(\mathbf{p},\mathbf{p}')$. □

The time complexity of Algorithm 1 is $O((d-\lambda)d)$.

4.3 Skew-Based Bound Functions

We illustrate the idea behind our bound functions in Fig. 4.

- First, we transform histograms $\mathbf{q}$ and $\mathbf{p}$ into sparser histograms $\mathbf{q}'$ and $\mathbf{p}'$. To control their sparsity, we introduce a parameter λ in the transform operation and require that $\Phi(\mathbf{q}') = \Phi(\mathbf{p}') = \lambda$.
- Then, we derive lower and upper bound functions for $emd(\mathbf{q},\mathbf{p})$ by using the transformed histograms (i.e., $\mathbf{q}',\mathbf{p}'$) and their relationships with the original histograms (i.e., $\mathbf{q},\mathbf{p}$).

As shown in Fig. 4, our bounds for $emd_c(\mathbf{q},\mathbf{p})$ depend on three terms $emd_c(\mathbf{q},\mathbf{q}')$, $emd_c(\mathbf{p},\mathbf{p}')$, and $emd_c(\mathbf{q}',\mathbf{p}')$. Since $\mathbf{q}'$ and $\mathbf{p}'$ are sparse, we can compute $emd_c(\mathbf{q}',\mathbf{p}')$ efficiently by the idea in Section 4.1. However, this idea cannot be used to accelerate the computation of $emd_c(\mathbf{q},\mathbf{q}')$, and $emd_c(\mathbf{p},\mathbf{p}')$. To reduce the computation time, we replace $emd_c(\mathbf{q},\mathbf{q}')$, $emd_c(\mathbf{p},\mathbf{p}')$ by fast-to-compute upper bounds $UB(\mathbf{q},\mathbf{q}'), UB(\mathbf{p},\mathbf{p}')$ (cf. Section 4.2). Specifically, we propose the following parametric functions in terms of λ and call them as *skew-based bound functions*

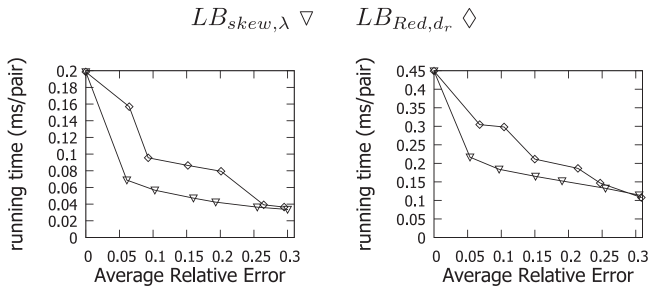


Fig. 5. Case study for our $LB_{skew,\lambda}$ and LB_{Red,d_r} [42].

$$LB_{skew,\lambda}(\mathbf{q},\mathbf{p}) = emd_c(\mathbf{q}',\mathbf{p}') - UB(\mathbf{q},\mathbf{q}') - UB(\mathbf{p},\mathbf{p}')$$
$$UB_{skew,\lambda}(\mathbf{q},\mathbf{p}) = emd_c(\mathbf{q}',\mathbf{p}') + UB(\mathbf{q},\mathbf{q}') + UB(\mathbf{p},\mathbf{p}'), \quad (2)$$

such that (i) $\Phi(\mathbf{q}') = \Phi(\mathbf{p}') = \lambda$ and (ii) $UB(\cdot,\cdot)$ is an upper bound function of $emd_c(\cdot,\cdot)$.

Observe that both $LB_{skew,\lambda}(\mathbf{q},\mathbf{p})$ and $UB_{skew,\lambda}(\mathbf{q},\mathbf{p})$ share all the terms, suggesting an opportunity for shared computation for both bounds.

Lemma 3 shows that $LB_{skew,\lambda}(\mathbf{q},\mathbf{p})$ and $UB_{skew,\lambda}(\mathbf{q},\mathbf{p})$ are lower and upper bound functions for $emd_c(\mathbf{q},\mathbf{p})$, respectively.

Lemma 3. *For any histograms $\mathbf{q},\mathbf{p}$, we have: $LB_{skew,\lambda}(\mathbf{q},\mathbf{p}) \le emd_c(\mathbf{q},\mathbf{p}) \le UB_{skew,\lambda}(\mathbf{q},\mathbf{p})$.*

Proof. By the triangle inequality (Lemma 1), we obtain

$$emd_c(\mathbf{q},\mathbf{p}') \le emd_c(\mathbf{q},\mathbf{q}') + emd_c(\mathbf{q}',\mathbf{p}')$$
$$emd_c(\mathbf{q},\mathbf{p}) \le emd_c(\mathbf{q},\mathbf{p}') + emd_c(\mathbf{p}',\mathbf{p}).$$

Adding these two inequalities, we get

$$\begin{aligned}emd_c(\mathbf{q},\mathbf{p}) &\le emd_c(\mathbf{q}',\mathbf{p}') + emd_c(\mathbf{q},\mathbf{q}') + emd_c(\mathbf{p}',\mathbf{p})\\ &\le emd_c(\mathbf{q}',\mathbf{p}') + UB(\mathbf{q},\mathbf{q}') + UB(\mathbf{p},\mathbf{p}').\end{aligned}$$

This implies that $emd_c(\mathbf{q},\mathbf{p}) \le UB_{skew,\lambda}(\mathbf{q},\mathbf{p})$.

By using Lemma 1 in another way, we obtain

$$emd_c(\mathbf{q}',\mathbf{p}) \ge emd_c(\mathbf{q}',\mathbf{p}') - emd_c(\mathbf{p}',\mathbf{p})$$
$$emd_c(\mathbf{q},\mathbf{p}) \ge emd_c(\mathbf{q}',\mathbf{p}) - emd_c(\mathbf{q}',\mathbf{q}).$$

Adding these two inequalities, we get

$$\begin{aligned}emd_c(\mathbf{q},\mathbf{p}) &\ge emd_c(\mathbf{q}',\mathbf{p}') - emd_c(\mathbf{q}',\mathbf{q}) - emd_c(\mathbf{p}',\mathbf{p})\\ &\ge emd_c(\mathbf{q}',\mathbf{p}') - UB(\mathbf{q},\mathbf{q}') - UB(\mathbf{p},\mathbf{p}').\end{aligned}$$

This implies that $emd_c(\mathbf{q},\mathbf{p}) \ge LB_{skew,\lambda}(\mathbf{q},\mathbf{p})$. □

Regarding the time complexity, the transformation operation (in Section 4.2) takes $O((d-\lambda)d)$ time, and the exact EMD computation on transformed histograms takes $O(\lambda^3 \log \lambda)$ time. Thus, the total time complexity is: $O((d-\lambda)d + \lambda^3 \log \lambda)$.

4.4 Case Study on Parametric Lower Bound Functions

Recall from Table 2, LB_{Red,d_r} [42] is the only parametric lower bound function in the literature. In order to compare the effectiveness of our parametric bound functions, we first sample 1,000 $(\mathbf{q},\mathbf{p})$ pairs from CAL-RGB and CAL-Lab datasets (see Section 6.1.1 for details). Then we test the running time (per pair) with respective to the average relative error of each bound, i.e., $\frac{1}{1000}\sum_{(\mathbf{q},\mathbf{p})} E_{\mathbf{q},\mathbf{p}}(LB_{Red,d_r}(\mathbf{q},\mathbf{p}))$ and $\frac{1}{1000}\sum_{(\mathbf{q},\mathbf{p})} E_{\mathbf{q},\mathbf{p}}(LB_{skew,\lambda}(\mathbf{q},\mathbf{p}))$. We select the most suitable parameters of λ and d_r such that the relative errors to be approximately 0.05, 0.1, 0.15, 0.2, 0.25 and 0.3. As shown in Fig. 5, our bound function $LB_{skew,\lambda}$ generally outperforms the existing bound function LB_{Red,d_r} under the small average relative error (e.g., 0 to 0.2). The main reason is our bound function $LB_{skew,\lambda}$ applies the smallest bin value shift first greedy strategy (cf. Fig. 3) which is more suitable for skewed data. However, LB_{Red,d_r} does not consider this property.

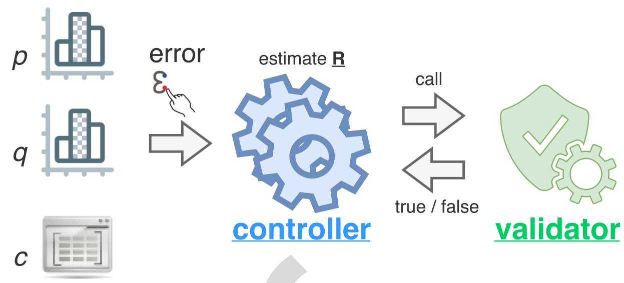


Fig. 6. Framework.

5 Approximation Framework

Although the lower/upper bound functions (cf. Table 2 and Section 4) may be used to compute approximate EMD value R, they provide no guarantee on the relative error (i.e., $\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) \le \epsilon$).

In contrast, we propose a framework to compute an approximate EMD value with bounded error. Our framework leverages on lower/upper bound functions for EMD. As shown in Fig. 6, our framework consists of the following two components.

- The *controller* selects a lower bound function and an upper bound function. Then it computes a lower bound ℓ, an upper bound u, and an approximate result R which is the value between ℓ and u.
- The *validator* receives information (e.g., ℓ, u, R) from the controller, and then checks whether the relative error definitely satisfies $\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) \le \epsilon$.

If the validator returns true, then the controller reports R to the user. Otherwise, the controller needs to obtain tighter bounds for ℓ and u, and repeats the above procedure.

5.1 Validator

In order to secure the correctness of our framework, we specify the following requirements for the validator:

- If it returns true, then it guarantees that the approximate result R must satisfy $\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) \le \epsilon$.
- Otherwise, it does not provide any guarantee for R.

For brevity in the remaining section, we use ℓ, u and e to represent $LB(\mathbf{q},\mathbf{p})$, $UB(\mathbf{q},\mathbf{p})$ and $emd_c(\mathbf{q},\mathbf{p})$.

As shown in Fig. 7, since the validator does not know the exact value e, it cannot directly compute the relative error of R, i.e., $\mathbb{E}_{\mathbf{q},\mathbf{p}}(R)$. Nevertheless, the validator can compute the *maximum possible* relative error of R, according to Lemma 4.

Lemma 4. $\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) \le \max(\frac{R}{\ell} - 1, 1 - \frac{R}{u})$.

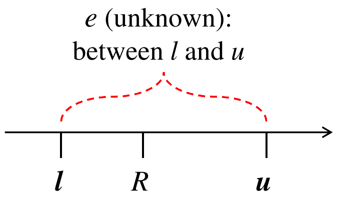


Fig. 7. Validation.

Proof. By the definition of $\mathbb{E}_{\mathbf{q},\mathbf{p}}(R)$, we have

$$\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) = \frac{|R-e|}{e} = \left|\frac{R}{e} - 1\right|.$$

Case 1: $R \geq e$

$$\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) = \frac{R}{e} - 1 \leq \frac{R}{\ell} - 1 \quad (\text{By } e \geq \ell).$$

Case 2: $R < e$

$$\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) = 1 - \frac{R}{e} \leq 1 - \frac{R}{u} \quad (\text{By } e \leq u).$$

Combining both cases, we obtain the following inequality:

$$\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) \leq \max\left(\frac{R}{\ell} - 1, 1 - \frac{R}{u}\right).$$

□

Our next step is to find the optimal R in order to minimize the maximum possible relative error $\max(\frac{R}{\ell} - 1, 1 - \frac{R}{u})$. According to Theorem 1, it is minimized when $R = \frac{2\ell u}{\ell+u}$, and the maximum possible relative error becomes $\frac{u-\ell}{u+\ell}$.

In subsequent sections, we use the notation $\mathbb{E}_{max}(\ell, u)$ to represent $\frac{u-\ell}{u+\ell}$.

Theorem 1. *If* $R = \frac{2\ell u}{\ell+u}$, *then:*

1) $\max(\frac{R}{\ell} - 1, 1 - \frac{R}{u})$ *achieves minimum.*
2) $\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) \leq \frac{u-\ell}{u+\ell}$.

Proof. For (1), we observe that the first term $\frac{R}{\ell} - 1$ is monotonic increasing with R and the second term $1 - \frac{R}{u}$ is monotonic decreasing with R. In order to minimize it, we set

$$\frac{R}{\ell} - 1 = 1 - \frac{R}{u} \iff R = \frac{2\ell u}{\ell + u}.$$

For (2),

$$\begin{aligned}\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) &= \frac{|R-e|}{e} = \left|\frac{R}{e} - 1\right| = \left|\frac{2\ell u}{e(\ell+u)} - 1\right| \\ &\leq \max\left(\left|\frac{2\ell u}{\ell(\ell+u)} - 1\right|, \left|\frac{2\ell u}{u(\ell+u)} - 1\right|\right) \\ &= \frac{u-\ell}{u+\ell}.\end{aligned}$$

□

Therefore, once the condition $\frac{u-\ell}{u+\ell} \leq \epsilon$ is fulfilled, $R = \frac{2\ell u}{\ell+u}$ can achieve the bounded error $E_{\mathbf{q},\mathbf{p}}(R) \leq \epsilon$.

5.2 Training-Free Controllers

In this section, we propose two control algorithms: Adaptive (ADA) and Lightweight Adaptive (ADA-L) which utilize our developed parametric lower and upper bound functions (cf. Section 4). These two control methods can be readily used on-the-fly because they do not need any training.

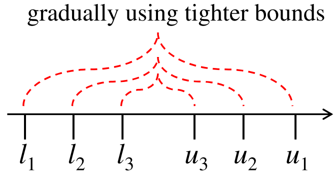


Fig. 8. Adaptive approach.

5.2.1 *Adaptive* (ADA)

Recall from Section 4.3, our parametric dual bound functions $LB_{skew,\lambda}$ and $UB_{skew,\lambda}$ depend on the parameter λ, which can affect both the running time and the error. This calls for an automatic method for selecting a suitable value for λ, upon the arrival of the $(\mathbf{q}, \mathbf{p})$-pair.

We propose the adaptive approach (ADA) as illustrated in Fig. 8. It gradually applies tighter bounds until passing the validation test. We present this method in Algorithm 2. It consists of an adaptive phase which performs validation by our parametric bound functions $LB_{skew,\lambda}, UB_{skew,\lambda}$ in ascending order of λ. We denote the increasing sequence of λ values by Λ (cf. Line 2). The algorithm executes our parametric bound functions in ascending order of $\lambda \in \Lambda$ until passing. It terminates as soon as it passes the validation test.

Algorithm 2. Adaptive Algorithm (ADA)

```
1: procedure ADA (histogram q, histogram p, cost matrix c,
   error threshold ϵ)
2:   initialize the sequence Λ of increasing integers
3:   for each λ ∈ Λ do                ▷ compute LB_skew,λ, UB_skew,λ
4:     (q′, ub_1) ← Skew-Transform (q, λ)
5:     (p′, ub_2) ← Skew-Transform (p, λ)
6:     temp ← emd_c(q′, p′)                       ▷ expensive call
7:     ℓ ← temp − ub_1 − ub_2; u ← temp + ub_1 + ub_2
8:     if E_max(ℓ, u) ≤ ϵ then                       ▷ Theorem 1
9:       return R = 2ℓu/(ℓ+u)
10:  return emd_c(q, p)                              ▷ expensive call
```

We propose one instantiation for Λ below:

- *Exponential sequence:* We introduce a parameter $\alpha > 1$ and construct $\Lambda = \langle \lfloor \alpha^i \rfloor : i \geq 0, \lfloor \alpha^i \rfloor < d \rangle$. For example, when $\alpha = 1.4$ and $d = 25$, the sequence is: $\langle 1, 1, 1, 2, 3, 5, 7, 10, 14, 20 \rangle$. In implementation, we omit duplicate integers in the sequence.

Theoretically, we show that ADA can be worse than the ADA-Opt (which knows the optimal λ value in advance for each $(\mathbf{q}, \mathbf{p})$ pair) by only a constant factor 5.18 if $\alpha = 1.2$.

Lemma 5. *For every* $(\mathbf{q}, \mathbf{p})$ *pair, let* $\mathbb{T}(\text{ADA}(\mathbf{q}, \mathbf{p}))$ *and* $\mathbb{T}(\text{ADA-Opt}(\mathbf{q}, \mathbf{p}))$ *be the running time of* ADA$(\mathbf{q}, \mathbf{p})$ *and* ADA-Opt$(\mathbf{q}, \mathbf{p})$ *respectively. If* $\alpha = 1.2$, *we have*

$$\frac{\mathbb{T}(\text{ADA}(\mathbf{q}, \mathbf{p}))}{\mathbb{T}(\text{ADA-Opt}(\mathbf{q}, \mathbf{p}))} \leq 5.18.$$

The detailed proof is shown in Appendix (Section 8).

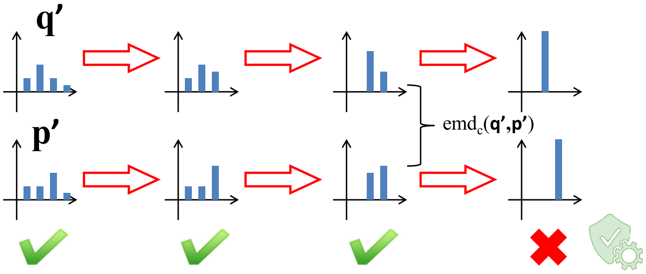


Fig. 9. Lightweight adaptive approach.

5.2.2 Lightweight Adaptive (ADA-L)

ADA may examine several λ and compute $emd_c(\mathbf{q}', \mathbf{p}')$ multiple times (in the adaptive phase). Thus, ADA can be expensive when ϵ is small. To avoid such overhead, we propose a lightweight version of the adaptive method, called ADA-L, such that it computes $emd_c(\mathbf{q}', \mathbf{p}')$ exactly once. Even though ADA-L does not have theoretical performance guarantee as ADA (cf. Lemma 5), the practical efficiency performance, as will be shown in experimental section, is better than ADA.

We show this ADA-L method in Fig. 9. This algorithm (cf. Algorithm 3) applies the skew-transform operation on histograms $\mathbf{q}'$ and $\mathbf{p}'$ such that they have one more zero bin. If the validation condition is satisfied, then we continue the loop. Otherwise, we terminate the loop and return $emd_c(\mathbf{q}', \mathbf{p}')$ as the approximate result.

The correctness of the validation condition is established by the following theorem.

Theorem 2. *If $R = emd_c(\mathbf{q}', \mathbf{p}')$, then*

$$\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) \leq \frac{UB(\mathbf{q}, \mathbf{q}') + UB(\mathbf{p}, \mathbf{p}')}{LB(\mathbf{q}, \mathbf{p})}. \tag{3}$$

Proof. In the proof of Lemma 3, we have: $emd_c(\mathbf{q}, \mathbf{p}) \leq emd_c(\mathbf{q}', \mathbf{p}') + UB(\mathbf{q}, \mathbf{q}') + UB(\mathbf{p}, \mathbf{p}')$ and $emd_c(\mathbf{q}, \mathbf{p}) \geq emd_c(\mathbf{q}', \mathbf{p}') - UB(\mathbf{q}, \mathbf{q}') - UB(\mathbf{p}, \mathbf{p}')$. Thus, we obtain: $|emd_c(\mathbf{q}', \mathbf{p}') - emd_c(\mathbf{q}, \mathbf{p})| \leq UB(\mathbf{q}, \mathbf{q}') + UB(\mathbf{p}, \mathbf{p}')$.

$$\begin{aligned}
\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) &= \frac{|emd_c(\mathbf{q}', \mathbf{p}') - emd_c(\mathbf{q}, \mathbf{p})|}{emd_c(\mathbf{q}, \mathbf{p})} \quad \text{(Given value of } R\text{)} \\
&\leq \frac{UB(\mathbf{q}, \mathbf{q}') + UB(\mathbf{p}, \mathbf{p}')}{emd_c(\mathbf{q}, \mathbf{p})} \\
&\leq \frac{UB(\mathbf{q}, \mathbf{q}') + UB(\mathbf{p}, \mathbf{p}')}{LB(\mathbf{q}, \mathbf{p})} \quad \text{(By Definition 1).}
\end{aligned}$$

□

Therefore, once the condition $UB(\mathbf{q}, \mathbf{q}') + UB(\mathbf{p}, \mathbf{p}') \leq \epsilon LB(\mathbf{q}, \mathbf{q}')$ is fulfilled, $R = emd_c(\mathbf{q}', \mathbf{p}')$ can achieve the bounded error $\mathbb{E}_{\mathbf{q},\mathbf{p}}(R) \leq \epsilon$.

Before computing R, we can bound the error $\mathbb{E}_{\mathbf{q},\mathbf{p}}(R)$ by using three terms, namely $UB(\mathbf{q}, \mathbf{q}')$, $UB(\mathbf{p}, \mathbf{p}')$ and $LB(\mathbf{q}, \mathbf{p})$. In the algorithm, the value ℓ (cf. Line 2) corresponds to $LB(\mathbf{q}, \mathbf{p})$, and the value $ub_{sum} + ub_1 + ub_2$ (cf. Line 8) corresponds to $UB(\mathbf{q}, \mathbf{q}') + UB(\mathbf{p}, \mathbf{p}')$.

The most appealing property of this ADA-L is that it avoids the expensive call of EMD operation in each iteration. The fast incremental upper bound function (cf. Lemma 2) incrementally updates $UB(\mathbf{q}, \mathbf{q}')$ and $UB(\mathbf{p}, \mathbf{p}')$ which leads to efficient computation in each iteration.

Algorithm 3. Lightweight Adaptive Method (ADA-L)

1: **procedure** ADA-L (histogram $\mathbf{q}$, histogram $\mathbf{p}$, cost matrix c, error threshold ϵ)
2: $\quad \ell \leftarrow LB_{Proj}(\mathbf{q}, \mathbf{p})$ ▷ Use the fastest lower bound function
3: $\quad ub_{sum} \leftarrow 0$
4: $\quad \mathbf{q}' \leftarrow \mathbf{q}, \mathbf{p}' \leftarrow \mathbf{p}$
5: $\quad$ **while** $\Phi(\mathbf{q}') > 1$ and $\Phi(\mathbf{p}') > 1$ **do**
6: $\quad\quad (\mathbf{q}'', ub_1) \leftarrow$ Skew-Transform $(\mathbf{q}', c, \Phi(\mathbf{q}') - 1)$
7: $\quad\quad (\mathbf{p}'', ub_2) \leftarrow$ Skew-Transform $(\mathbf{p}', c, \Phi(\mathbf{p}') - 1)$
8: $\quad\quad$ **if** $ub_{sum} + ub_1 + ub_2 \leq \epsilon \cdot \ell$ **then** ▷ Theorem 2
9: $\quad\quad\quad ub_{sum} \leftarrow ub_{sum} + ub_1 + ub_2$
10: $\quad\quad\quad \mathbf{q}' \leftarrow \mathbf{q}'', \mathbf{p}' \leftarrow \mathbf{p}''$
11: $\quad\quad$ **else**
12: $\quad\quad\quad$ break
13: $\quad$ return $emd_c(\mathbf{q}', \mathbf{p}')$ ▷ expensive call

5.3 Training-Based Controller (ADA-H)

Some applications, e.g., image retrieval [22], [23], [33] and image classification [45], might have huge historical workload data. Such rich information can help to pick the bounds such that the framework can find a good approximate result R at lower cost, compared to training-free controllers (e.g., ADA-L).

As discussed in Table 2, there exist several lower bound functions $LB \in Set_{LB}$ and upper bound functions $UB \in Set_{UB}$ at lower cost as compared to our ADA-L. If we can select the fast combination of LB and UB for $(\mathbf{q}, \mathbf{p})$ with the validation condition $E_{max}(\ell, u) \leq \epsilon$ is fulfilled, then the response time can be further reduced. The question is which bound functions in Set_{LB} and Set_{UB} should be picked.

In this work, we propose another control method ADA-H which picks the sequence of bound functions (from Set_{LB} and Set_{UB}) based on ϵ and the historical workload Γ in the offline training stage. After that, the chosen sequence of bounds will be used to handle the online computation.

5.3.1 Offline Training Stage

This stage requires historical workload data, which is defined as the collection Γ of pairs $(\mathbf{q}, \mathbf{p})$. We first build the following tables for Γ.

Definition 3 (V_ϵ-Table). *$V_\epsilon(LB, UB)$ denotes the set of $(\mathbf{q}, \mathbf{p})$ pairs where their estimated results (using LB and UB) pass the validation stage subject to the error threshold ϵ.*

$$V_\epsilon(LB, UB) = \{(\mathbf{q}, \mathbf{p}) \in \Gamma | E_{max}(LB(\mathbf{q}, \mathbf{p}), UB(\mathbf{q}, \mathbf{p})) \leq \epsilon\}.$$

Definition 4 ($\mathbb{T}$-Table). *$\mathbb{T}$-Table records the response time $\mathbb{T}(LB(\mathbf{q}, \mathbf{p}), UB(\mathbf{q}, \mathbf{p}))$ of different bound functions $LB \in Set_{LB}$ and $UB \in Set_{UB}$ for every $(\mathbf{q}, \mathbf{p})$ pair.*

Given the V_ϵ-Table and the $\mathbb{T}$-Table of a workload set Γ, we want to find a sequence of bounds (from Set_{LB} and Set_{UB}) such that the response time of evaluating these bounds is minimized subject to a constraint that all estimated result R of $(\mathbf{q}, \mathbf{p}) \in \Gamma$ satisfies the validation condition $E_{\mathbf{q},\mathbf{p}}(R) \leq \epsilon$.

Our idea, as shown in Fig. 10, is to iteratively partition the workload Γ into subsets by using a sequence of validations. We denote Γ_i and F_i to be the set of remaining and filtered pairs after the ith validation. A *feasible* sequence of bounds in this example is $\{(lb_j, ub_i), (lb_y, ub_x), \ldots\}$. These

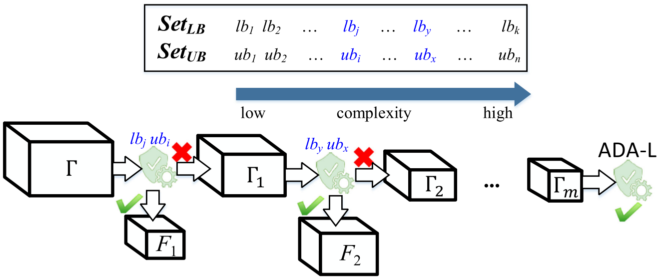


Fig. 10. Picking a sequence of bounds in the offline training stage.

bounds secure that every pair in Γ fulfills the error threshold ϵ, which is verified by the validator (based on the information in V_ϵ-Table). Note that we need ADA-L at the last step to secure the feasibility (since ADA-L is an adaptive method so it always finds a result fulfilling $E_{\mathbf{q},\mathbf{p}}(R) \le \epsilon$).

Among all feasible sequences of bounds, we want to find the best sequence of bounds that minimizes the response time (based on the information in $\mathbb{T}$-Table). However, finding the best sequence of bounds that optimizes the objective and fulfills the constraint is a combinatorial problem. For the sake of processing, we simplify the problem and use a greedy method to find the sequence of bounds.

1) We sort the bounds of Set_{LB} and Set_{UB} based on their running time.
2) Pick the fastest pair of bounds into the suggested sequence S and call the validator to partition Γ into Γ_1 and F_1.
3) Get the next pair (LB, UB) of bounds and call the validator to partition Γ_i into Γ_{i+1} and F_{i+1} in which the estimated response time (cf. Eq. (4)) is minimized.

$$\sum_{(\mathbf{q},\mathbf{p})\in\Gamma_i} \mathbb{T}(LB(\mathbf{q},\mathbf{p}), UB(\mathbf{q},\mathbf{p})) + \sum_{(\mathbf{q},\mathbf{p})\in\Gamma_{i+1}} \mathbb{T}(\text{ADA-L}(\mathbf{q},\mathbf{p})). \tag{4}$$

4) If Eq. (4) is smaller than $\sum_{(\mathbf{q},\mathbf{p})\in\Gamma_i} \mathbb{T}(\text{ADA-L}(\mathbf{q},\mathbf{p}))$, we pick this pair of bounds into the suggested sequence S. Otherwise, we choose ADA-L into S, then report S as the final sequence and terminate this algorithm.
5) Repeat Step (3) until the Γ_i becomes $\emptyset$.

5.3.2 Online Stage

When processing a new pair $(\mathbf{q}, \mathbf{p})$ (of the same application domain), our controller only evaluates those picked sequence of bounds in S (cf. Algorithm 4). This chosen sequence S offers very good performance in practice since the validator skips to check many ineffective bound pairs. We will show the detailed performance in the experimental section.

Algorithm 4. ADA-H (Online)

```
procedure ADA-H ONLINE(histogram q, histogram p,
    sequence of bounds S, cost matrix c, error threshold ε)
  for each (LB, UB) ∈ S do
    ℓ ← LB(q, p), u ← UB(q, p)
    if E_max(ℓ, u) ≤ ε then                    ▷ Theorem 1
      return R = 2ℓu/(ℓ+u)
  Return ADA-L(q, p, c, ε)
```

TABLE 3
Raw Datasets of Images

Raw dataset	# of images	Used in
UW	1,109	[12]
VOC	5,011	[13]
COR (Corel)	10,800	[41]
CAL (Caltech)	30,609	[14]
FL (Flickr)	1,000,000	[18]

TABLE 4
Methods for Extracting Color Histograms

Histogram name	Dimensionality	Used in
RGB	64	[12]
Lab	256	[33]

6 Experimental Evaluation

We introduce the experimental setting in Section 6.1. Then, we evaluate the effectiveness of different bound functions in Section 6.2. Next, we present the experiments for approximate EMD computation in Section 6.3. Then, we demonstrate the effectiveness and the efficiency of our methods on k-NN content-based image retrieval in Section 6.4. We implemented all algorithms in C++ and conducted experiments on an Intel i7 3.4 GHz PC running Ubuntu.

6.1 Experimental Setting

6.1.1 Datasets

We have collected five raw datasets of images as listed in Table 3. These datasets have been used extensively in the computer vision and the information retrieval areas. For each raw image, we apply a *histogram extraction method* to obtain a color histogram $\mathbf{p}$.

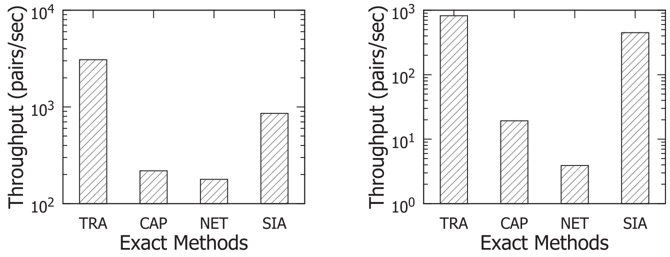
We consider two representative methods for extracting color histograms, as shown in Table 4. RGB color histogram is the traditional representation method since 1990s [15], [37], [38]. It is still effective for content-based image retrieval [12] and EMD-based applications [46]. Lab color histogram is extensively used in computer vision and image retrieval applications [30], [33], [36], [43]. We follow the setting of [12], [33] to extract these two types of color histograms. Specifically, we divide the color space uniformly into $4 \times 4 \times 4$ partitions and $4 \times 8 \times 8$ partitions, for RGB and Lab respectively. According to [33], [39], we compute the cost matrix c by setting $c_{i,j}$ to the Euclidean distance between the centers of partitions i and j in the color space.

By using each histogram extraction method on each raw dataset, we obtain ten datasets: UW-RGB, VOC-RGB, COR-RGB, CAL-RGB, FL-RGB, UW-Lab, VOC-Lab, COR-Lab, CAL-Lab, FL-Lab. We name each dataset by the format [raw dataset]-[histogram name].

6.1.2 Exact EMD Computation

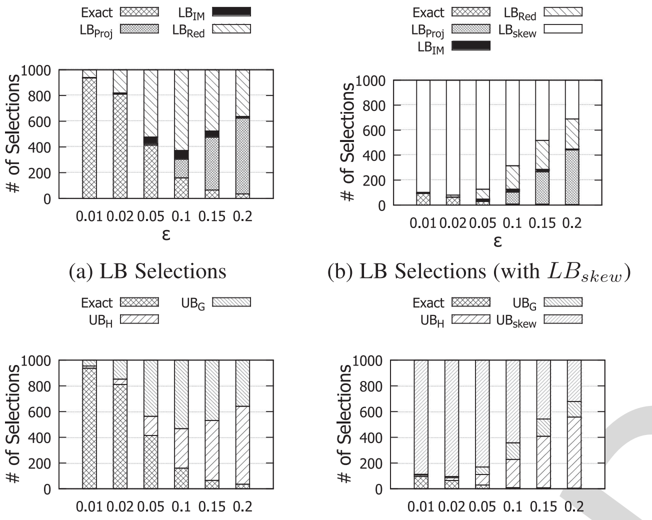
For the sake of fairness, we consider representative methods for computing exact EMD and attempt to identify the fastest one on our datasets. These methods include: (i) two algorithms CAP and NET from the Lemon Graph Library,[1] (ii) SIA [39] and (iii) TRA [33].

1. http://lemon.cs.elte.hu/trac/lemon



(a) CAL-RGB (b) CAL-Lab

Fig. 11. Throughput of exact EMD computation methods.



(a) LB Selections (b) LB Selections (with LB_{skew})

(c) UB Selections (d) UB Selections (with UB_{skew})

Fig. 12. Number of lower and upper bound functions selected by Oracle in CAL-Lab dataset.

In this experiment, we randomly sample 1000 pairs of histograms from a dataset, and measure the throughput (number of processed pairs/sec) of each method. Fig. 11 shows the throughput on two datasets: CAL-RGB and CAL-Lab. Observe that TRA performs the best on both datasets. We obtain similar trends on other datasets. Therefore, we use TRA for exact EMD computation in the remaining experimental study.

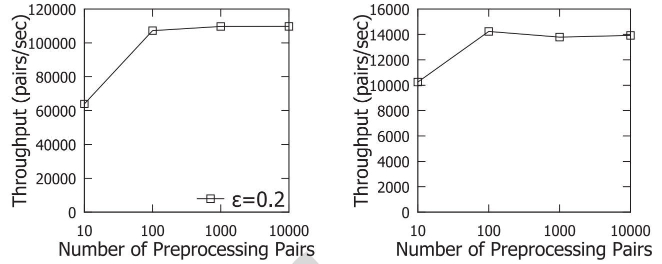
6.1.3 Oracle

In order to demonstrate the usefulness of different control algorithms in our approximation framework, we define the following omniscient method Oracle.

Definition 5. *Given histograms* $\mathbf{q}, \mathbf{p}$, *we define the* optimal *pair of lower and upper bound functions as follows:*

$$\begin{aligned}\mathsf{Oracle}(\mathbf{q},\mathbf{p}) = \underset{(LB,UB)}{\arg\min}\{&\mathbb{T}(LB(\mathbf{q},\mathbf{p}), UB(\mathbf{q},\mathbf{p})) \\ &: LB \in Set_{LB}, UB \in Set_{UB}, \\ &\mathbb{E}_{max}(LB(\mathbf{q},\mathbf{p}), UB(\mathbf{q},\mathbf{p})) \le \epsilon\}.\end{aligned} \tag{5}$$

For each $(\mathbf{q},\mathbf{p})$ pair, Oracle pre-knows the fastest pair of (ℓ, u) which fulfills the validation condition $E_{max}(\ell, u) \le \epsilon$. As such, it acts as the most efficient solution for all control methods in our approximation framework. In later sections,



(a) COR-RGB (b) COR-Lab

Fig. 13. Throughput versus number of pre-processing pairs in ADA-H, fixing $\epsilon = 0.2$.

we will demonstrate how efficient of our solutions compare with Oracle.

6.2 Are Parametric Dual Bound Functions Useful?

In this section, we compare the effectiveness of our derived parametric dual bound functions with existing bounds. Recall from Section 6.1.3, Oracle$(\mathbf{q},\mathbf{p})$ is denoted by the best lower and upper bound pair for $(\mathbf{q},\mathbf{p})$ pair. Therefore, if the bound is frequently selected by Oracle, that bound is more useful.

We first randomly sample 1,000 $(\mathbf{q},\mathbf{p})$ pairs of histogram from the dataset CAL-Lab. For a given error threshold ϵ, we count the number of lower and upper bound functions selected by Oracle in these histogram pairs. Observe from Figs. 12a and 12c, Exact are frequently chosen at small ϵ (e.g., 0.01 and 0.02). Therefore, existing bound functions are not useful for small ϵ case in which LB_{skew} and UB_{skew} are widely applicable for these cases (cf. Figs. 12b and 12d). Moreover, LB_{skew} and UB_{skew} are frequently selected compared with other bound functions in small to moderate ϵ values (0.01-0.2) by Oracle.

6.3 Approximate EMD Computation

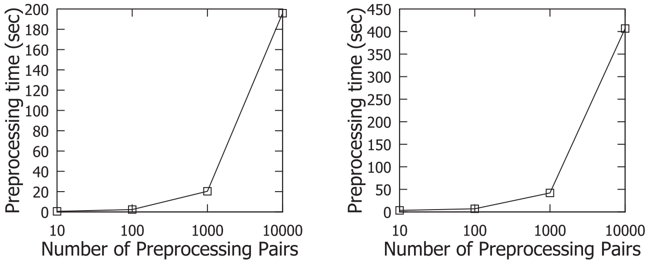
In this section, we test the throughput and the error of various approximate EMD computation methods. Our competitors are lower/upper bound functions LB_{IM}, LB_{Proj}, LB_{Red}, UB_G, UB_H [6], [20], [34], [39], [42] and two approximate methods in the computer vision and machine learning area, which are FEMD[2] [30] and Sinkhorn [3] respectively. Our proposed methods are ADA, ADA-L and ADA-H. Note that our methods offer guarantee on error threshold ϵ, but our competitors do not provide such guarantee. By default, we set $\epsilon = 0.2$.

In each dataset, we randomly sample 1,000 testing pairs of histograms, and measure the throughput (pairs/sec) of all methods.

6.3.1 Effect of Pre-Processing in ADA-H

The performance of our ADA-H method depends on the number of pairs in the pre-processing steps. For fairness, we make sure that pre-processing pairs are different from testing pairs. In this experiment, we vary the number of pre-processing pairs and plot the throughput in Fig. 13. Observe that the throughput becomes stable when the numbers of preprocessing pairs are 100, 1,000 and 10,000. By default, we use 100 pre-processing pairs for ADA-H in subsequent experiments.

2. Implementation at http://www.ariel.ac.il/sites/ofirpele/FastEMD/



(a) COR-RGB (b) COR-Lab

Fig. 14. Preprocessing time in ADA-H.

Fig. 14 shows the preprocessing time in COR-RGB and COR-Lab datasets. The preprocessing time is proportional to the number of pairs used for training. However, using 100 training-pairs leads to stable performance in the online stage, as shown in Fig. 13. Therefore, the training time is not the bottleneck in general.

6.3.2 Comparisons Among Our Methods

In order to conduct meaningful comparisons, we compare our methods with three benchmarks.

- Exact: the fastest exact EMD computation method (TRA), according to Fig. 11.
- ADA-Opt: an optimal skew method that knows the optimal λ value in advance. Its throughput serves as the upper bound of our skew-based methods ADA and ADA-L.
- Oracle: the theoretically optimal method, which knows the optimal pair of bound functions in advance (Section 6.1.3).

Fig. 15 plots the throughput of ADA-Opt and our adaptive methods (ADA and ADA-L). Observe that ADA-L can achieve a similar throughput compared to ADA-Opt. Since ADA-L performs better than ADA in practice, we exclude ADA for subsequent experiments.

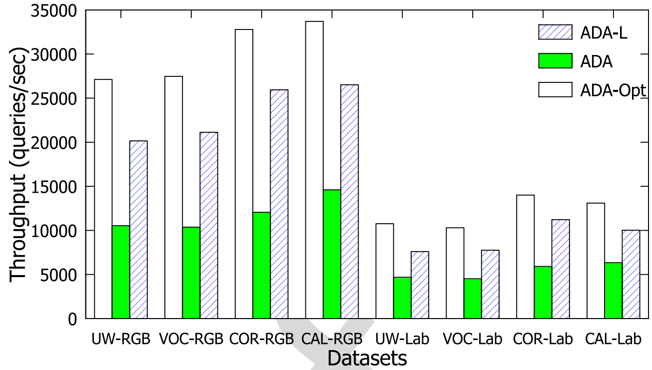
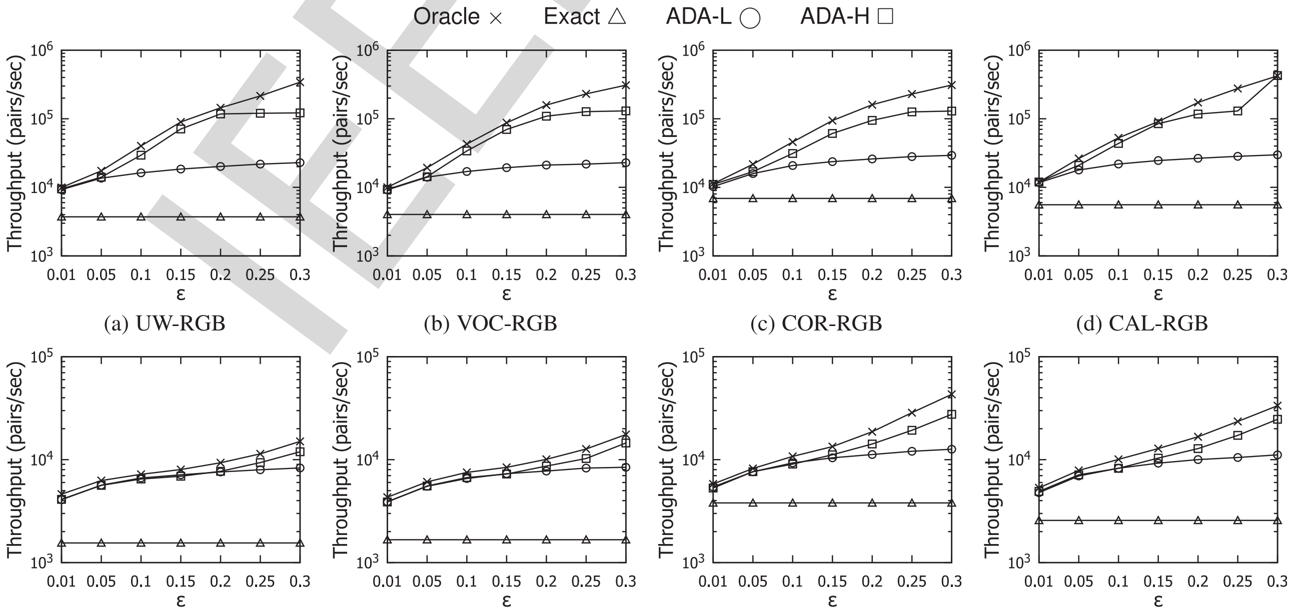


Fig. 15. Throughput between our methods and ADA-Opt method, fixing $\epsilon = 0.2$.

In Fig. 16, we study the effect of the error threshold ϵ on the throughput of our methods ADA-H and ADA-L. We also report the throughput of Oracle and Exact in this experiment. In general, our methods achieve higher throughput than Exact. Our best method ADA-H can achieve significant speed-up (e.g., by an order of magnitude) on various datasets. Even though Exact can also achieve 1,600-7,000 pairs/sec in all datasets, which are not slow in general, some applications, for example: kNN-image retrieval and classification [33] or EMD similarity join [17] involve many EMD computations, especially for large-scale datasets, which make Exact inefficient for these applications. We will discuss in detail in our case study (cf. Section 6.4).

In the next experiment, we vary the dimensionality d of the dataset by using RGB color histogram with $d = m^3$ bins. Fig. 17 shows the throughput of Exact, ADA-L and ADA-H as a function of the dimensionality. As expected, the throughput decreases when the dimensionality d increases. Our best method ADA-H consistently outperforms Exact by an order of magnitude.

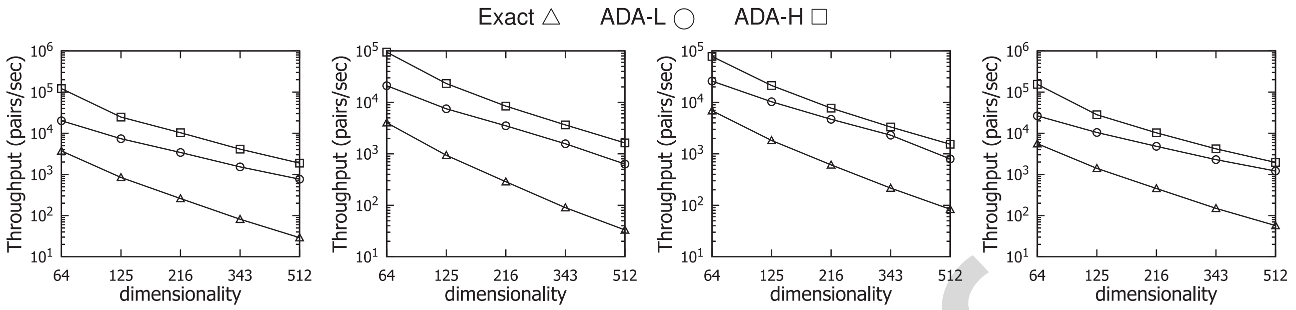
To demonstrate the stability of our best method ADA-H, we sort the response time (in increasing order) for all those 1,000 sample pairs in CAL-RGB and CAL-Lab datasets and



(a) UW-RGB (b) VOC-RGB (c) COR-RGB (d) CAL-RGB

(e) UW-Lab (f) VOC-Lab (g) COR-Lab (h) CAL-Lab

Fig. 16. Effect of the error threshold ϵ on different datasets.



(a) UW-RGB (b) VOC-RGB (c) COR-RGB (d) CAL-RGB

Fig. 17. Effect of the dimensionality d on different datasets.

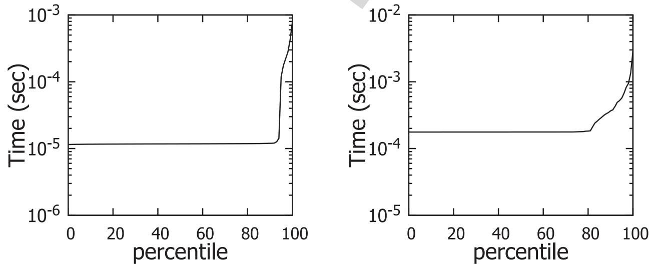
then plot the percentile statistics in Fig. 18. Observe that the response time is stable for the method ADA-H for 80 percent of sample pairs, whereas the remaining 20 percent of sample pairs may take longer time to evaluate.

6.3.3 Comparisons with Competitors

We proceed to compare our best method ADA-H with other approximation methods. We classify our competitors into two types:

- *non-parametric approximation methods* whose throughput cannot be tuned (i.e., LB_{IM}, LB_{Proj}, UB_G, UB_H [6], [20], [34], [39]),
- *parametric approximation methods* whose throughput can be tuned via a parameter (i.e., LB_{Red} [42], FEMD [30], SIA_B [39], Sinkhorn [3]).

Table 5 shows how we choose the parameter for each parametric approximation method. LB_{Red} [42] is the dimension reduction technique for EMD, we choose different reduced dimensions, d_{red} for conducting this experiment. FEMD [30] utilizes the threshold to truncate the edges (i, j) which costs c_{ij} exceed the threshold in the bipartite flow network of $EMD(\mathbf{q}, \mathbf{p})$. Tang et al. [39] develop the progressive lower bound function and apply UB_G for upper bound function, SIA_B combines these bounding functions with our approximation framework (cf. Section 5). Since SIA_B utilizes our approximation framework, this is the only existing parametric approximate method which can provide the relative error guarantee of the returned result. Altschuler et al. [3] utilize Sinkhorn iterative projection algorithm to obtain the approximate EMD value in which they provide the δ-additive error guarantee of the returned result. To provide reasonable setting of δ, we set seven values (cf. Table 5) for each $(\mathbf{q}, \mathbf{p})$ pair based on its own EMD value $e = emd_c(\mathbf{q}, \mathbf{p})$.



(a) CAL-RGB (b) CAL-Lab

Fig. 18. Response time (sec) versus percentile of ADA-H, fixing $\epsilon = 0.2$.

In order to obtain a holistic view, we plot the throughput and the error of a method as a point. The error of a method is taken as the average relative error (per tested pair). The performance of all methods are shown in Fig. 19.

First, we compare the performance of ADA-H with nonparametric approximation methods. Since LB_{Proj}, UB_H, LB_{IM} and UB_G take at most $O(d^2)$ time, they are normally faster than ADA-H (especially for the points with small error) but incur high error. Next, we consider the parametric approximation methods, ADA-H obtains better performance in terms of both throughput and error in most of the tested cases.

6.4 Case Study on kNN Content-Based Image Retrieval

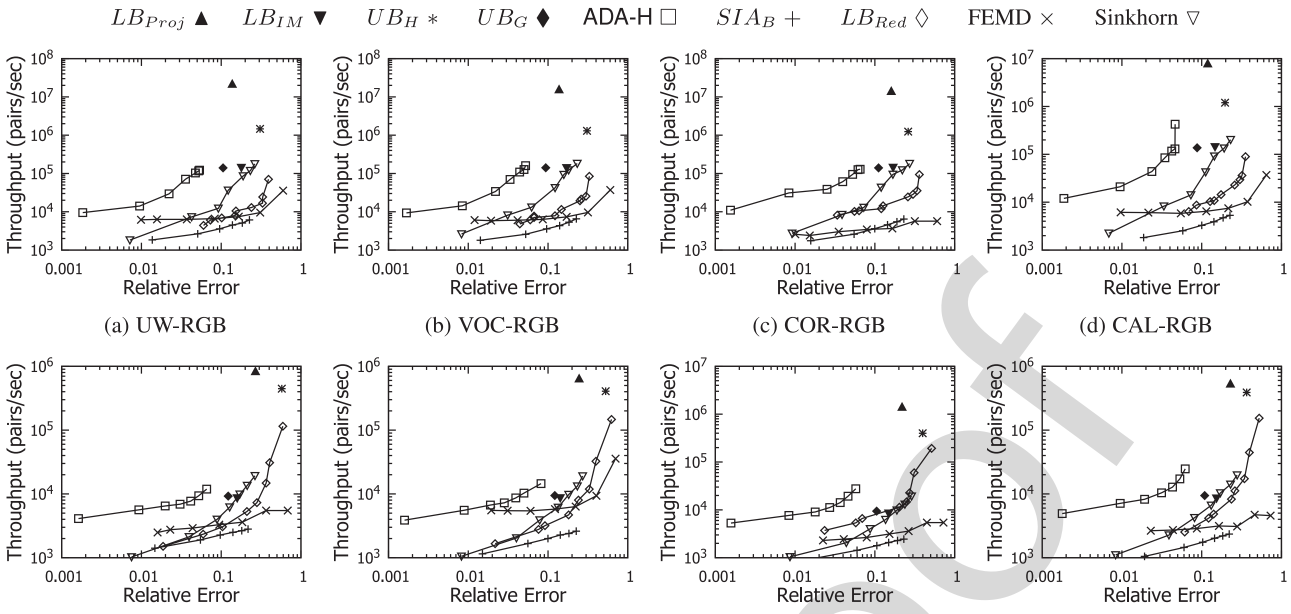
We conduct case study to demonstrate the effectiveness and the efficiency of methods on kNN content-based image retrieval. Our competitor, denoted by Exact-kNN, is the fastest known method for exact kNN search with EMD [39]. Our kNN search method is the same as Exact-kNN, except that we replace the refinement stage by our approximate method ADA-H.

We use the largest datasets (FL-RGB, FL-Lab) for testing. In each dataset, we randomly sample 100 query histograms. For each method, we measure its efficiency as the query throughput (queries/sec), and measure its effectiveness as the average precision per query, where the precision is defined as the fraction of the retrieved results in the exact kNN results.

We investigate the effect of ϵ on the kNN retrieval performance in terms of both precision and efficiency. In this experiment, we set $k = 100$ by default and vary ϵ from 0.05 to 0.3. Fig. 20 shows the precision and the throughput of ADA-H compared with Exact-kNN. Observe that the precision remains above 0.8 (in Figs. 20a and 20b) when ϵ is relatively large (e.g., $\epsilon = 0.3$). Figs. 20c and 20d demonstrate that ADA-H outperforms Exact-kNN by 3-5x and 3.5-7x on FL-RGB and FL-Lab, respectively.

TABLE 5
Parameter Tuning

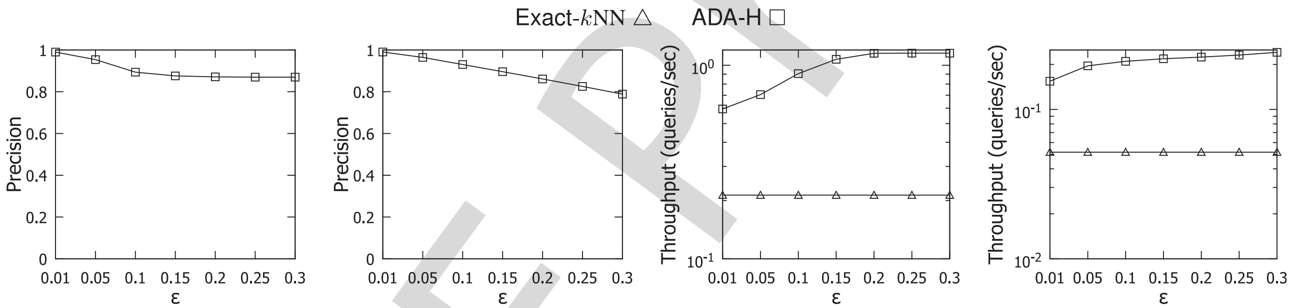
Method	para.	RGB	Lab
LB_{Red}	d_{red} [42]	{12,18,24,...,60}	{24,56,88,...,248}
FEMD	Threshold [30]	{50,100,150,...,350}	{12,24,36,...,84}
SIA_B, ADA-H	ϵ	{0.01,0.05,0.1,0.15,...,0.3}	
Sinkhorn	δ	$\{0.01\,e, 0.05e, 0.1e, 0.15e, \ldots, 0.3e\}$	



(a) UW-RGB (b) VOC-RGB (c) COR-RGB (d) CAL-RGB

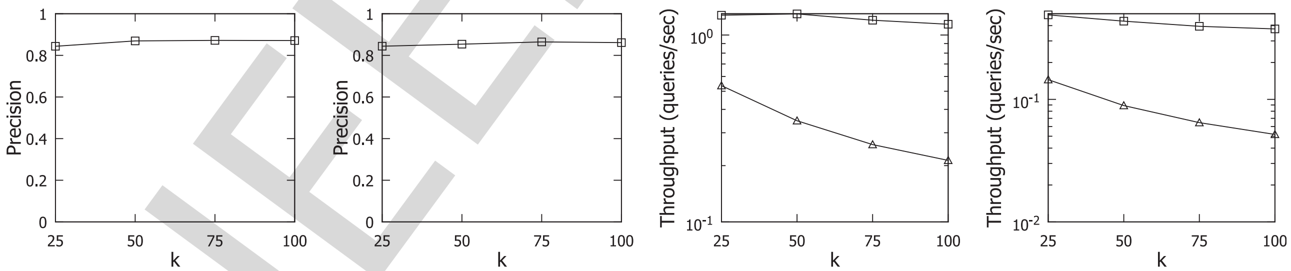
(e) UW-Lab (f) VOC-Lab (g) COR-Lab (h) CAL-Lab

Fig. 19. Comparisons with all approximation methods on different datasets.



(a) precision on FL-RGB (b) precision on FL-Lab (c) throughput on FL-RGB (d) throughput on FL-Lab

Fig. 20. Effect of the error threshold ϵ on the kNN content-based image retrieval, fixing $k = 100$.



(a) precision on FL-RGB (b) precision on FL-Lab (c) throughput on FL-RGB (d) throughput on FL-Lab

Fig. 21. Effect of the result size k on the performance of kNN content-based image retrieval, fixing $\epsilon = 0.2$.

Next, we test the effect of k on the kNN retrieval performance in terms of both precision and efficiency. Figs. 21a and 21b show the precision of ADA-H as a function of k. The precision of ADA-H is high and it is independent of k. In Figs. 21c and 21d, we observe that the throughput of ADA-H is not sensitive to k. On the other hand, the throughput of Exact-kNN is linearly proportional to k. Overall, ADA-H outperforms Exact-kNN by 2.38-5x and 3.38-7.26x on FL-RGB and FL-Lab, respectively.

7 Conclusion

This paper studies the computation of approximate EMD value with bounded error. Specifically, we guarantee to return an approximate EMD value that is within $1 \pm \epsilon$ times the exact EMD value. We have presented an adaptive approach for our problem. In our experimental evaluation, we have used five raw image datasets with two histogram extraction methods. Our best method, ADA-H, yields up to an order of magnitude speedup over the fastest exact computation algorithm. We have also evaluated the effectiveness of ADA-H on the kNN content-based image retrieval application. In the future, we plan to investigate how to extend our approximation framework to other applications, e.g., EMD similarity join, and other similarity functions, e.g., edit distance.

8 Appendix

8.1 The Exponential Sequence Λ in ADA

In the following, we compare the running time of ADA with an ADA-Opt algorithm, which knows additional information in advance. Specifically, ADA-Opt knows the best λ to be chosen for a given $(\mathbf{q},\mathbf{p})$-pair.

According to the analysis below, by using the exponential sequence with $\alpha = 1.2$, the running time of ADA is bounded by a constant multiple (i.e., 5.18) of the running time of ADA-Opt.

8.1.1 Analysis

For the sake of analysis, we model the running time as follows.

$$\mathbb{T}(emd_c(\mathbf{q},\mathbf{p})) = d^3 \log d \tag{6}$$

$$\mathbb{T}(LB_{skew,\lambda}(\mathbf{q},\mathbf{p}), UB_{skew,\lambda}(\mathbf{q},\mathbf{p})) = \lambda^3 \log \lambda, \tag{7}$$

because the state-of-the-art EMD computation algorithm requires $O(d^3 \log d)$ time. We fix the hidden constant factor to 1 and the log base to 2.

Our competitor is the ADA-Opt method, which knows in advance the value d_* as defined below

$$d_* = \min\{\lambda : \mathbb{E}_{max}(LB_{skew,\lambda}(\mathbf{q},\mathbf{p}), UB_{skew,\lambda}(\mathbf{q},\mathbf{p})) \le \epsilon\}. \tag{8}$$

Therefore, ADA-Opt suffices to call the fastest $LB_{skew,\lambda}$ and $UB_{skew,\lambda}$ once, then passes the validation test. Thus, we have: $\mathbb{T}(\text{ADA-Opt}(\mathbf{q},\mathbf{p})) = d_*^3 \log d_*$.

We assume that $\mathbb{E}_{max}(LB_{skew,\lambda}(\mathbf{q},\mathbf{p}), UB_{skew,\lambda}(\mathbf{q},\mathbf{p}))$ decreases when λ increases. Therefore, ADA terminates when $\lambda \in \Lambda$ is the smallest integer that satisfies $\lambda \ge d_*$.

We define the ratio of the running time of ADA to ADA-Opt

$$Ratio = \frac{\mathbb{T}(\text{ADA}(\mathbf{q},\mathbf{p}))}{\mathbb{T}(\text{ADA-Opt}(\mathbf{q},\mathbf{p}))}. \tag{9}$$

Theorem 3. *Given the exponential sequence* $\Lambda = \langle \lfloor \alpha^i \rfloor : i \ge 0, \lfloor \alpha^i \rfloor < d \rangle$, *we have*

$$\frac{\mathbb{T}(\text{ADA}(\mathbf{q},\mathbf{p}))}{\mathbb{T}(\text{ADA-Opt}(\mathbf{q},\mathbf{p}))} \le \frac{\alpha^6(1+\log_2\alpha)}{\alpha^3 - 1}.$$

Proof. When $d_* = 1$, the iteration $i = 0$ can directly handle it. We have $Ratio = 1$ in this case. In the remaining discussion, we assume that $d_* > 1$.

Let n be the positive number such that

$$\alpha^{n-1} < d_* \le \alpha^n. \tag{10}$$

ADA terminates when it reaches the iteration $n = \lceil \log_\alpha d_* \rceil$. Thus, we have

$$\mathbb{T}(\text{ADA}(\mathbf{q},\mathbf{p})) = \sum_{i=0}^{n} \lfloor \alpha^i \rfloor^3 \log \lfloor \alpha^i \rfloor \le \sum_{i=1}^{n} \alpha^{3i} \log \alpha^i$$

$$\begin{aligned} Ratio &\le \frac{\sum_{i=1}^{n} \alpha^{3i} \log \alpha^i}{{d_*}^3 \log d_*} \\ &\le \frac{n \log \alpha}{{d_*}^3 \log d_*} \cdot \sum_{i=1}^{n} \alpha^{3i} \\ &= \frac{n \log \alpha}{{d_*}^3 \log d_*} \cdot \frac{\alpha^3(\alpha^{3n}-1)}{\alpha^3 - 1}. \end{aligned}$$

Since $n = \lceil \log_\alpha d_* \rceil$, we have $n \le \log_\alpha d_* + 1$ and $\alpha^{3n} \le \alpha^{3(\log_\alpha d_* + 1)} = {d_*}^3\alpha^3$. Therefore

$$\begin{aligned} Ratio &\le \frac{(\log_\alpha d_* + 1)\log \alpha}{{d_*}^3 \log d_*} \cdot \frac{\alpha^3({d_*}^3\alpha^3 - 1)}{\alpha^3 - 1} \\ &= \frac{\alpha^3({d_*}^3\alpha^3 - 1)}{{d_*}^3(\alpha^3 - 1)} \cdot \left(\frac{\log d_*}{\log \alpha} + 1\right) \cdot \frac{\log \alpha}{\log d_*} \\ &= \frac{\alpha^3({d_*}^3\alpha^3 - 1)}{{d_*}^3(\alpha^3 - 1)} \cdot (1 + \log_{d_*}\alpha) \\ &= \frac{\alpha^3}{\alpha^3 - 1} \cdot \left(\alpha^3 - \frac{1}{{d_*}^3}\right)(1 + \log_{d_*}\alpha). \end{aligned}$$

Since $\log_{d_*}\alpha \le \log_2\alpha$ and $-\frac{1}{d_*} \le 0$, we have

$$Ratio \le \frac{\alpha^6(1+\log_2\alpha)}{\alpha^3 - 1}.$$

□

Corollary 1. *Given that* $\alpha = 1.2$, *we have*

$$\frac{\mathbb{T}(\text{ADA}(\mathbf{q},\mathbf{p}))}{\mathbb{T}(\text{ADA-Opt}(\mathbf{q},\mathbf{p}))} \le 5.18.$$

Proof. By finding the minimum value of $\frac{\alpha^6(1+\log_2\alpha)}{\alpha^3-1}$ with a numerical solver. □

Acknowledgments

This work was supported by grant GRF152201/14E from the Hong Kong RGC. Leong Hou U was supported by MYRG-2016-00182- FST from UMAC RC.

References

[1] P. K. Agarwal and R. Sharathkumar, "Approximation algorithms for bipartite matching with metric and geometric costs," in *Proc. 46th Annu. ACM Symp. Theory Comput.*, 2014, pp. 555–564.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows - Theory, Algorithms and Applications*. Upper Saddle River, NJ, USA: Prentice Hall, 1993.

[3] J. Altschuler, J. Weed, and P. Rigollet, "Near-linear time approximation algorithms for optimal transport via sinkhorn iteration," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1961–1971.

[4] A. Andoni, P. Indyk, and R. Krauthgamer, "Earth mover distance over high-dimensional spaces," in *Proc. 19th Annu. ACM-SIAM Symp. Discr. Algorithms*, 2008, pp. 343–352.

[5] D. Applegate, T. Dasu, S. Krishnan, and S. Urbanek, "Unsupervised clustering of multidimensional distributions using earth mover distance," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 636–644.

[6] I. Assent, A. Wenning, and T. Seidl, "Approximation techniques for indexing the earth mover's distance in multimedia databases," in *Proc. 22nd Int. Conf. Data Eng.*, 2006, pp. 11–11.

[7] I. Assent, M. Wichterich, T. Meisen, and T. Seidl, "Efficient similarity search using the earth mover's distance for large multimedia databases," in *Proc. IEEE 24th Int. Conf. Data Eng.*, 2008, pp. 307–316.

[8] J. Blanchet, A. Jambulapati, C. Kent, and A. Sidford, "Towards optimal running times for optimal transport," *CoRR*, 2018. [Online]. Available: https://arxiv.org/abs/1810.07717

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[10] M. H. Coen, M. H. Ansari, and N. Fillmore, "Comparing clusterings in space," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 231–238.

[11] S. D. Cohen and L. J. Guibas, "The earth mover's distance: Lower bounds and invariance under translation technical report," Stanford Univ., Stanford, CA, Rep. no. STAN-CS-TR-97-1597, 1997.

[12] T. Deselaers, D. Keysers, and H. Ney, "Features for image retrieval: An experimental comparison," *Inf. Retrieval*, vol. 11, no. 2, pp. 77–107, 2008.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge 2007 (VOC2007) Results," 2007. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

[14] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Tech. Rep. 7694, Caltech, 2007. [Online]. Available: https://authors.library.caltech.edu/7694/

[15] J. L. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 7, pp. 729–736, Jul. 1995.

[16] J. Huang, R. Zhang, R. Buyya, and J. Chen, "MELODY-JOIN: Efficient earth mover's distance similarity joins using MapReduce," in *Proc. IEEE 30th Int. Conf. Data Eng.*, 2014, pp. 808–819.

[17] J. Huang, R. Zhang, R. Buyya, J. Chen, and Y. Wu, "Heads-join: Efficient earth mover's distance similarity joins on hadoop," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 6, pp. 1660–1673, Jun. 2016.

[18] M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Proc. 1st ACM Int. Conf. Multimedia Inf. Retrieval*, 2008, pp. 39–43.

[19] P. Indyk, "A near linear time constant factor approximation for Euclidean bichromatic matching (cost)," in *Proc. 18th Annu. ACM-SIAM Symp. Discr. Algorithms*, 2007, pp. 39–42.

[20] M. Jang, S. Kim, C. Faloutsos, and S. Park, "A linear-time approximation of the earth mover's distance," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 505–514.

[21] M. Jang, S. Kim, C. Faloutsos, and S. Park, "Accurate approximation of the earth mover's distance in linear time," *J. Comput. Sci. Technol.*, vol. 29, no. 1, pp. 142–154, 2014.

[22] F. Jing, M. Li, H. Zhang, and B. Zhang, "An efficient and effective region-based image retrieval framework," *IEEE Trans. Image Process.*, vol. 13, no. 5, pp. 699–709, May 2004.

[23] F. Jing, M. Li, H. Zhang, and B. Zhang, "Relevance feedback in region-based image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 672–681, May 2004.

[24] M. Kapralov and R. Panigrahy, "NNS lower bounds via metric expansion for l_∞ and EMD," in *Proc. Int. Colloquium Autom. Lang. Program.*, 2012, pp. 545–556.

[25] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger, "From word embeddings to document distances," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 957–966.

[26] V. Ljosa, A. Bhattacharya, and A. K. Singh, "Indexing spatially sensitive distance measures using multi-resolution lower bounds," in *Proc. Int. Conf. Extending Database Technol.*, 2006, pp. 865–883.

[27] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, 2005.

[28] G. Nikolentzos, P. Meladianos, and M. Vazirgiannis, "Matching node embeddings for graph similarity," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2429–2435.

[29] J. B. Orlin, "A faster strongly polynominal minimum cost flow algorithm," in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 377–387.

[30] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 460–467.

[31] K. Quanrud, "Approximating optimal transport with linear programs," in *Proc. 2nd Symp. Simplicity Algorithms*, 2019, pp. 6:1–6:9.

[32] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proc. 6th Int. Conf. Comput. Vis.*, 1998, pp. 59–66.

[33] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.

[34] B. E. Ruttenberg and A. K. Singh, "Indexing the earth mover's distance using normal distributions," *Proc. VLDB Endowment*, vol. 5, no. 3, pp. 205–216, 2011.

[35] J. Sherman, "Generalized preconditioning and undirected minimum-cost flow," in *Proc. 28th Annu. ACM-SIAM Symp. Discr. Algorithms*, 2017, pp. 772–780.

[36] S. Shirdhonkar and D. W. Jacobs, "Approximate earth mover's distance in linear time," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[37] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. C. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[38] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, 1991.

[39] Y. Tang, L. H. U, Y. Cai, N. Mamoulis, and R. Cheng, "Earth mover's distance based similarity search at scale," *Proc. VLDB Endowment*, vol. 7, no. 4, pp. 313–324, 2013.

[40] F. Wang and L. J. Guibas, "Supervised earth mover's distance learning and its computer vision applications," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 442–455.

[41] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-sensitive integrated matching for picture libraries," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 9, pp. 947–963, Sep. 2001.

[42] M. Wichterich, I. Assent, P. Kranen, and T. Seidl, "Efficient EMD-based similarity search in multimedia databases via flexible dimensionality reduction," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 199–212.

[43] J. Xu, B. Lei, Y. Gu, M. Winslett, G. Yu, and Z. Zhang, "Efficient similarity join based on earth mover's distance using MapReduce," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 8, pp. 2148–2162, Aug. 2015.

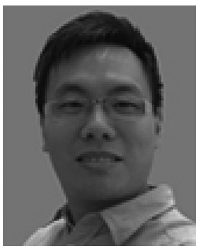
[44] J. Xu, Z. Zhang, A. K. H. Tung, and G. Yu, "Efficient and effective similarity search over probabilistic data based on earth mover's distance," *Proc. VLDB Endowment*, vol. 3, no. 1, pp. 758–769, 2010.

[45] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *Int. J. Comput. Vis.*, vol. 73, no. 2, pp. 213–238, 2007.

[46] Q. Zhao, Z. Yang, and H. Tao, "Differential earth mover's distance with its applications to visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 274–287, Feb. 2010.

Tsz Nam Chan received the bachelor's degree in electronic and information engineering and the PhD degree in computer science from Hong Kong Polytechnic University, in 2014 and 2019, respectively. He is currently a research associate with the University of Hong Kong. His research interests include multidimensional similarity search, pattern matching, and kernel methods for machine learning.

Man Lung Yiu received the bachelor's degree in computer engineering and the PhD degree in computer science from the University of Hong Kong, in 2002 and 2006, respectively. Prior to his current post, he worked at Aalborg University for three years starting in the Fall of 2006. He is now an associate professor with the Department of Computing, The Hong Kong Polytechnic University. His research focuses on the management of complex data, in particular query processing topics on spatiotemporal data and multidimensional data.



Leong Hou U received the BSc degree in computer science and information engineering from Taiwan Chi Nan University, the MSc degree in e-commerce from the University of Macau, and the PhD degree in computer science from the University of Hong Kong. He is now an associate professor with the State Key Laboratory of Internet of Things for Smart City and the Department of Computer and Information Science, University of Macau. His research interests include spatial and spatio-temporal databases, advanced query processing, crowdsourced query processing, information retrieval, data mining, and optimization problems. He is a member of the IEEE.