All reviewers: We thank all reviewers for their positive feedback. We are encouraged that they find our work offers *a significant algorithmic breakthrough* (R1), *addresses an important open problem* (R1), *works very well in practice* (R2), and is *interesting* (R2, R3) and *promising* (R2, R4). ▶ *Generality.* Let us clarify that our algorithm and Theorem 3 are valid for many existing UOT problems. If one wants to compute the Kantrovich-Rubinstein (KR) distance, our algorithm computes it in $O(n \log^2 n)$ time by setting $\lambda_c = \lambda_d = \lambda$, and Theorem 3 provides an approximation guarantee. Note that any existing algorithms for the KR distance require at least $O(n^2)$ time. Likewise, if one wants to compute the Figalli's distance, our algorithm computes it efficiently by setting $\lambda_c(x) = \lambda_d(x) = d(x, \partial\mathcal{X})$. See also related work.

Reviewer #1: ▶ *Two hyper-parameters.* When two hyper-parameters are troublesome, we recommend to use the KR distance or optimal partial transport, which has only one hyper-parameter and can be computed efficiently by our algorithm. ▶ *Chicago Crime.* We compare the distributions of crime locations. For example, in some festival days, many crimes may happen in specific locations, and the number of crimes may suddenly increase/decrease. We can detect anomalies and clusters. ▶ *NY Taxi.* Existing methods for UOT are missing because *no existing methods can handle this dataset due to scalability*. Our method is the first UOT method that can handle million-scale datasets.

Reviewer #2: ▶ *Weight function.* The cost is the ground distance between the centers of regions of the quadtree. We will further clarify this in the camera-ready. ▶ *Metric Axiom.* Intuitively, when no mass is created or destructed, GKR is reduced to the standard OT, thus metric. When some mass is created or destructed, GKR is positive. Hence, $\mathrm{GKR}(\mu, \nu) = 0$ iff $\mu = \nu$ almost everywhere (under positivity conditions for λ_c and λ_d). We will formally state this.

Reviewer #3: ▶ *(1) How generality is reflected.* Many existing OT problems are obtained by setting λ_c and λ_d appropriately. ▶ *(4) Recent methods.* The tree-sliced Wasserstein we used in the NY taxi dataset and Appendix E was published in NeurIPS 2019. That is a state-of-the art (standard, not unbalanced) OT method applicable to million-scale datasets. We also used the generalized Shinkhorn published in 2018 in the additional experiments below.

Reviewer #4: ▶ *(1) Hard to follow.* Our algorithm computes the GKR distance from leaf to root recursively. Figure 1 shows examples. Note that when we compute the transport in a parent node, the optimal assignments in the children subtrees are already computed recursively. When we merge two children, the dynamic programming determines the optimal transition (i.e., the optimal amount of mass that are transported to the left and right children). The proposed fast convolution algorithm speeds up this merge operation. We will provide more detailed descriptions and intuitions in the camera-ready. Furthermore, we will provide an open-sourced toolkit of our algorithm at GitHub. We believe that it will benefit many practitioners thanks to its fast computation. ▶ *(2) Beyond 2 dimensions.* The quadtree we used in the paper is *NOT* restricted to two dimensions. See [38, 31] for details. When the dimensions are high, clustering-based trees can be used [38]. To show this, we conduct additional experiments. First, we compute GKR for the Chicago Crime dataset with the additional time axis, using the *quadtree*. Each mass represents a crime in the 3-dimensional (longitude, latitude, time) space. We normalize each dimension so that each dimension has the same scale. Next, we compute GKR for the (unbalanced) Word Mover's Distance of the Twitter dataset [38] using *clustering-based trees*. Each measure represents a sentence, and each mass represents a word embedded in a 300-dimensional space computed by a pre-trained language model. We compare our algorithm with the groundtruth GKR distance in the Euclidean space, as we did in the main paper. Figure 2 shows that our algorithm can compute high dimensional GKR. ▶ *(3) No intersection case.* Our algorithm *is* feasible even if there is no intersections. In that case, each leaf node contains only the source or target mass. ▶ *(4) LP formulation.* We reported the accuracy in Figure 4 in the original paper. There, we used *exact* computation for the Euclidean GKR using an LP-like solver. Specifically, we used a network flow algorithm, which solves OT problems exactly (i.e., match exactly with the LP solution) and is faster than general-purpose LP solvers. We will clarify it. ▶ *Proof of Thm.2.* There, we consider the case where $\lambda \leq \delta/2 (\leq c/2)$ (See L.505). See also the definitions of δ and c in L.503-504. The opposite case (i.e., $\lambda < \delta/2$) is discussed in L.506-508. ▶ *Is OT_{tree} the same as $|v(P) - v(Q)|_1$ in [31]?* Exactly.

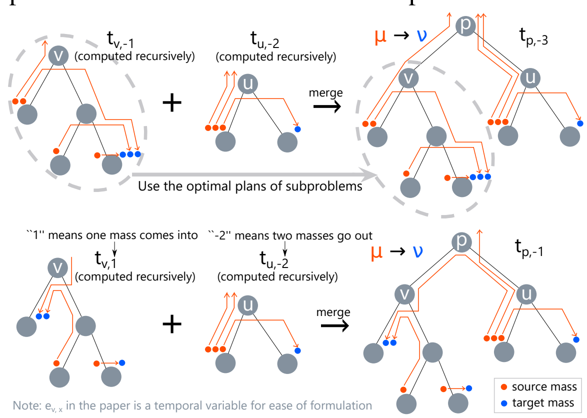

Figure 1: Two examples of DP computation.

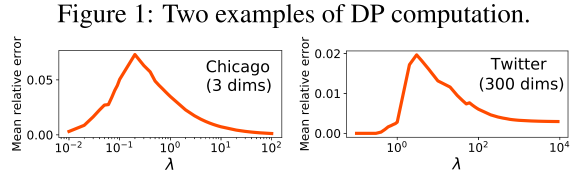

Figure 2: Accuracy in high dimensional cases.

Additional Experiments: We conducted experiments for the generalized Sinkhorn [16] with the same setting as Appendix E. We observed a similar tendency (k=0: 0.83, k=16: 0.37) to Tree GKR. The complete results are deferred to the camera-ready due to space limitation. Since the generalized Sinkhorn requires at least $O(n^2)$ time, its applicability is limited to thousand-scale datasets. Our algorithm is applicable to million-scale datasets keeping its performance. We also conducted document classification using Twitter dataset [38] and found that GKR improved the performance over the Word Mover's Distance (Accuracy: $0.719 \to 0.729$). The detailed results will be included in the camera-ready.