

STABILIZED SPARSE SCALING ALGORITHMS FOR ENTROPY REGULARIZED TRANSPORT PROBLEMS

BERNHARD SCHMITZER

Abstract. Scaling algorithms for entropic transport-type problems have become a very popular numerical method, encompassing Wasserstein barycenters, multi-marginal problems, gradient flows and unbalanced transport. However, a standard implementation of the scaling algorithm has several numerical limitations: the scaling factors diverge and convergence becomes impractically slow as the entropy regularization approaches zero. Moreover, handling the dense kernel matrix becomes unfeasible for large problems. To address this, we combine several modifications: A log-domain stabilized formulation, the well-known ε -scaling heuristic, an adaptive truncation of the kernel and a coarse-to-fine scheme. This permits the solution of larger problems with smaller regularization and negligible truncation error. A new convergence analysis of the Sinkhorn algorithm is developed, working towards a better understanding of ε -scaling. Numerical examples illustrate efficiency and versatility of the modified algorithm.

1. Introduction.

1.1. Motivation and Related Work.

Applications of Optimal Transport. Optimal transport (OT) is a classical optimization problem dating back to the seminal work of Monge and Kantorovich (see monographs [47, 39] for introduction and historical context). The induced Wasserstein distances lift a metric from a ‘base’ space (X, d) to probability measures over X . This is a powerful analytical tool, for example to study PDEs as gradient flows in Wasserstein space [24, 3]. With the increase of computational resources, OT has also become a popular numerical tool in image processing, computer vision and machine learning (e.g. [38, 42, 32, 18, 20]).

Many ideas have been presented to extend Wasserstein distances to general non-negative measures. We refer to [26, 13, 31, 15] and references therein for some context. A transport-type distance for general multi-channel signals is proposed in [46].

Computational Optimal Transport. To this day, the computational effort to solve OT problems remains the principal bottleneck in many applications. In particular large problems, or even multi-marginal problems, remain challenging both in terms of runtime and memory demand.

For the linear assignment problem and discrete transport problems there are (combinatorial) algorithms based on the finite dimensional linear programming formulation by Kantorovich, such as the Hungarian method [28], the auction algorithm [9], the network simplex [2] and more [22]. Typically, they work for (almost) arbitrary cost functions, but do not scale well for large, dense problems. On the other hand, there are more geometric solvers, relying on the polar decomposition [11], that tend to be more efficient. There is the famous fluid dynamic formulation by Benamou and Brenier [5], explicit computation of the polar decomposition [23], semi-discrete solvers [34, 30], and solvers of the Monge-Ampère equation [8, 7] among many others. However, these only work on very specific cost functions, most notably the squared Euclidean distance. In a compromise between efficiency and flexibility, several discrete coarse-to-fine solvers have been proposed that adaptively select sparse sub-problems [41, 35, 40].

Entropy Regularization for Optimal Transport. In [27] entropy regularization of the linear assignment problem is considered to allow application of smooth optimization techniques or the Sinkhorn matrix scaling algorithm [44]. For sufficiently small regularization the true optimal assignment can be extracted from the approximate solution. For increased numerical stability, the Sinkhorn algorithm is also reformulated in the log-domain. Similarly, in [17] the Sinkhorn algorithm is applied to solve an entropy regularized approximation of the discrete optimal transport problem. It is demonstrated that for moderate regularization strengths the algorithm is trivial to parallelize, easy to implement on GPUs and fast. Besides, it is shown that moderate regularization can actually be beneficial for classification applications. Regularization also makes the optimization problem more well-behaved (e.g. uniqueness of optimal coupling, optimal objective differentiable as function of marginal distributions), which led to the first practical numerical method for approximate computation of Wasserstein barycenters [19]. Today, this approach is widely used, for instance [45, 37, 46, 33].

More recently, the Sinkhorn algorithm has been extended to more general transport-type problems, such as multi-marginal problems and direct computation of Wasserstein barycenters [6], gradient flows [36] and unbalanced transport problems [14], resulting in a family of Sinkhorn-like diagonal scaling algorithms.

Convergence of the discrete regularized problem towards the unregularized limit is studied in [16].

In a continuous setting, this is related to the Schrödinger problem and the lazy gas experiment (see [29] for a review and a very general convergence proof). [12] provides a simpler and direct analysis for the 2-Wasserstein distance on \mathbb{R}^d and studies the limit of entropy regularized gradient flows.

Convergence Speed of Sinkhorn Algorithm. In [21] the convergence rate of the Sinkhorn algorithm is studied for positive kernel matrices, yielding a global linear convergence rate of the marginals in terms of Hilbert’s projective metric. However, applied to entropy regularized optimal transport, the contraction factor tends to one exponentially, as the regularization approaches zero. Thus, running the algorithm with this particular measure of convergence is often not practically feasible. In [25] the local convergence rate of the Sinkhorn algorithm near the solution is examined, based on a linearization of the iterations. This bound is tighter and more accurately describes the behaviour of the algorithm close to convergence. But these estimates do not apply when one starts far from the optimal solution, which is the usual case for small regularization parameters. In [27] a comparison is made between the Sinkhorn algorithm and the auction algorithm. In particular the role of the entropy regularization parameter is related to the slack parameter ε of the auction algorithm and it is pointed out that convergence of both algorithms becomes slower, as these parameters approach zero (but small parameters are required for good approximate solutions). For the auction algorithm this can provably be remedied by ε -scaling, where the ε parameter is gradually decreased during optimization. Analogously, it is suggested to gradually decrease entropy regularization during the Sinkhorn algorithm to accelerate optimization. Consequently, in the following we will also refer to the entropy regularization parameter as ε and to the gradual reduction scheme as ε -scaling. The ideas of [27] are refined in [43]. In particular, the latter proves convergence of a modified algorithm with ‘deformed iterations’ where ε is gradually decreased during the iterations, similar to ε -scaling. They show that the primal iterate converges to the unregularized solution if the decrease is sufficiently slow. Unfortunately, the number of iterations to reach a given value of ε increases exponentially, as ε decreases. Thus it is “mostly interesting from the theoretical point of view” [43, p. 8].

Limitations of Entropic Transport. Despite its considerable merits, there are some fundamental constraints to the naive entropy regularization approach. Entropy introduces some blur in the optimal assignment. While this may sometimes be beneficial (see above), in many applications it is considered a nuisance (e.g. it quickly smears distinct features in gradient flows), and one would like to run the scaling algorithm with as little regularization as possible. However, a standard implementation has some major numerical limitations, becoming increasingly severe as the regularization approaches zero. The diagonal scaling factors diverge in the limit of vanishing regularization, leading to numerical overflow and instabilities. Moreover, the algorithm requires an increasing number of iterations to converge. In practice this can often be remedied by ε -scaling, but its efficiency is not yet well understood theoretically. Therefore, numerically this limit is difficult to reach. In addition, naively storing the dense kernel matrix requires just as much memory as storing the full cost matrix in standard linear programming solvers and multiplications with the kernel matrix become increasingly slow. Thus, effective heuristics to avoid storing of, and multiplication by, the dense kernel matrix have been conceived, such as efficient Gaussian convolutions or approximation by a pre-factored heat kernel [45]. However, these remedies only work for particular (although relevant) problems, and do not solve the issues of blur and diverging scaling factors.

1.2. Contribution and Outline. In Section 2 we recall the framework for transport-type problems and corresponding scaling algorithms for their entropy regularized counterparts, as put forward in [14]. The main contributions of this article are twofold: In Section 3 we propose to combine four modifications of the Sinkhorn algorithm to address issues with numerical instability, slow convergence and large kernel matrices. In Section 4 a new convergence analysis for the Sinkhorn algorithm is derived, based on an analogy to the auction algorithm. The two sections can be read independently from each other. The modifications used in Section 3 are:

- *Section 3.1:* A log-domain stabilization of the Sinkhorn algorithm, as described in [14]. It allows to numerically run the algorithm at small regularizations while largely retaining the simple matrix scaling structure.
- *Section 3.2:* The well-known ε -scaling heuristic, to reduce the number of required iterations.
- *Section 3.3:* Sparsification of the kernel matrix by adaptive truncation, to reduce memory demand and accelerate iterations. We quantify the error induced by truncation and propose a truncation scheme which reliably yields small error bounds that are easy to evaluate. While truncation has

been proposed elsewhere (e.g. [33]), to the best of our knowledge the present article gives the first concrete bounds for the inflicted error.

- *Section 3.4*: A multi-scale scheme, inspired, for instance, by [41, 40, 35]. This serves two purposes: First, it allows for a more efficient computation of the truncated kernel. Second, we propose to combine the coarse-to-fine approach with simultaneous ε -scaling, which drastically reduces the number of variables during early stages of ε -scaling, without losing significant precision.

We emphasize that each modification builds on the previous ones (see Remark 10) and only combining all four leads to an algorithm that can solve large problems with significantly less runtime, memory and regularization, as compared to the naive algorithm. The adaptations extend to the more general scaling algorithms for transport-type problems presented in [14].

In Section 4 we develop a new convergence analysis of the Sinkhorn algorithm, based on analogy to the auction algorithm, different from the Hilbert metric approach of [21]. The structure of Section 4 is:

- *Section 4.1*: The classical auction algorithm for the linear assignment problem is recalled.
- *Section 4.2*: A slightly modified asymmetric variant of the Sinkhorn algorithm is given and a bound is derived for the number of iterations until a prescribed accuracy is reached. As for the auction algorithm, for fixed ε the maximal number of iterations scales as $\mathcal{O}(1/\varepsilon)$. This is in good agreement with numerical experiments (cf. Section 5.2). To avoid the difficulties with slow convergence in Hilbert’s projective metric (cf. Section 1.1) we choose a weaker, but reasonable, measure of convergence (cf. Remark 5).
- *Section 4.3*: We prove stability of optimal dual solutions of entropy regularized OT under changes of the regularization parameter. This also implies stability of dual solutions in the limit of vanishing regularization and therefore complements results of [16] (see also Remark 7).
- *Section 4.4*: Our eventual goal is a better theoretical understanding of the ε -scaling heuristic and its efficiency. We show that the above stability result is an important step and discuss missing steps for a full proof. To our knowledge (with the exception of [43], see above), these are the first theoretical results towards ε -scaling for the Sinkhorn algorithm.

Numerical experiments confirm the efficiency of the modified algorithm (Section 5.2). Examples for unbalanced optimal transport, barycenters, and Wasserstein gradient flows illustrate that the modified algorithms retain the versatility of the diagonal scaling algorithms presented in [6, 36, 14] (Section 5.3).

1.3. Notation and Preliminaries. We assume that the reader has a basic knowledge of convex optimization, such as convex conjugation, Fenchel–Rockafellar duality and primal-dual gaps (cf. [4]).

Throughout this article, we will consider transport problems between two discrete finite spaces X and Y . For a discrete, finite space Z (typically X , Y or $X \times Y$) we identify functions and measures over Z with vectors in $\mathbb{R}^{|Z|}$, which we simply denote by \mathbb{R}^Z . For $v \in \mathbb{R}^Z$, $z \in Z$ we write $v(z)$ for the component of v corresponding to z (subscript notation is reserved for other purposes). The standard Euclidean inner product is denoted by $\langle \cdot, \cdot \rangle$. The sets of vectors with positive and strictly positive entries are denoted by \mathbb{R}_+^Z and \mathbb{R}_{++}^Z . The probability simplex over Z is denoted by $\mathcal{P}(Z)$. We write $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$ for the extended real line and $\overline{\mathbb{R}}^Z$ for the space of vectors with possibly infinite components.

For $a, b \in \mathbb{R}^Z$ the operators \odot and \oslash denote pointwise multiplication and division, e.g. $a \odot b \in \mathbb{R}^Z$, $(a \odot b)(z) := a(z) \cdot b(z)$ for $z \in Z$. The functions \exp and \log are extended to \mathbb{R}^Z by pointwise application to all components: $\exp(a)(z) := \exp(a(z))$. We write $a \geq b$ if $a(z) \geq b(z)$ for all $z \in Z$, $a \geq 0$ if $a(z) \geq 0$ for all $z \in Z$ (and likewise for \leq , $>$ and $<$). For $a \in \mathbb{R}$, a_Z denotes the vector in \mathbb{R}^Z with all entries being a . We write $\max a$ and $\min a$ for the maximal and minimal entry of a .

For $\mu \in \mathbb{R}^Z$ and a subset $A \subset Z$ we also use the notation $\mu(A) := \sum_{z \in A} \mu(z)$, analogous to measures. We say $\mu \in \mathbb{R}^Z$ is absolutely continuous w.r.t. $\nu \in \mathbb{R}_+^Z$ and write $\mu \ll \nu$ when $[\nu(z) = 0] \Rightarrow [\mu(z) = 0]$. This is the discrete special case of absolute continuity for measures. The set $\text{spt } \mu := \{z \in Z : \mu(z) \neq 0\}$ is called support of μ . The power set of Z is denoted by 2^Z .

For a subset $\mathcal{C} \subset \mathbb{R}^Z$ the indicator function of \mathcal{C} over \mathbb{R}^Z is given by $\iota_{\mathcal{C}}(v) = 0$ if $v \in \mathcal{C}$ and $+\infty$ else. In particular, for $v, w \in \mathbb{R}^Z$ one finds $\iota_{\{v\}}(w) = 0$ if $v = w$ and $+\infty$ otherwise. Moreover, we merely write ι_+ for $\iota_{\mathbb{R}_+^Z}$. For $v \in \mathbb{R}^Z$ we introduce the short notation $\iota_{\leq v} : \mathbb{R}^Z \rightarrow \overline{\mathbb{R}}$ with $\iota_{\leq v}(w) = 0$ if $w(z) \leq v(z)$ for all $z \in Z$ and $+\infty$ otherwise.

The projection matrices $P_X \in \mathbb{R}^{X \times (X \times Y)}$ and $P_Y \in \mathbb{R}^{Y \times (X \times Y)}$ are given by

$$P_X(x, (x', y')) := \begin{cases} 1 & \text{if } x = x', \\ 0 & \text{else.} \end{cases}, \quad P_Y(y, (x', y')) := \begin{cases} 1 & \text{if } y = y', \\ 0 & \text{else.} \end{cases}$$

They act on some $\pi \in \mathbb{R}^{X \times Y}$ as follows:

$$(P_X \pi)(x) = \sum_{y \in Y} \pi(x, y) = \pi(\{x\} \times Y), \quad (P_Y \pi)(y) = \sum_{x \in X} \pi(x, y) = \pi(X \times \{y\}).$$

That is, they give the X and Y marginal in the sense of measures. Conversely, for some $v \in \mathbb{R}^X$, $w \in \mathbb{R}^Y$ we find $(P_X^\top v)(x, y) = v(x)$ and $(P_Y^\top w)(x, y) = w(y)$.

DEFINITION 1 (Kullback–Leibler Divergence). For $\mu, \nu \in \mathbb{R}^Z$ the Kullback–Leibler divergence of μ w.r.t. ν is given by

$$(1.1) \quad \text{KL}(\mu|\nu) := \begin{cases} \sum_{\substack{z \in Z: \\ \mu(z) > 0}} \mu(z) \log \left(\frac{\mu(z)}{\nu(z)} \right) - \mu(Z) + \nu(Z) & \text{if } \mu, \nu \geq 0, \mu \ll \nu, \\ +\infty & \text{else.} \end{cases}$$

The convex conjugate w.r.t. the first argument is given by $\text{KL}^*(\alpha|\nu) = \sum_{z \in Z} (\exp(\alpha(z)) - 1) \cdot \nu(z)$. The KL divergence plays a central role in this article and is used on various different base spaces. Sometimes, when referring to the KL divergence on a space Z , we will add a subscript KL_Z for clarification.

DEFINITION 2 (KL Proximal Step). For a convex, lower semicontinuous function $f : \mathbb{R}^Z \rightarrow \overline{\mathbb{R}}$ and a step size $\tau > 0$ the proximal step operator for the Kullback–Leibler divergence is given by

$$(1.2) \quad \text{prox}_{1/\tau} f : \mathbb{R}^Z \rightarrow \mathbb{R}^Z, \quad \mu \mapsto \underset{\nu \in \mathbb{R}^Z}{\text{argmin}} \left(\frac{1}{\tau} \text{KL}(\nu|\mu) + f(\nu) \right).$$

A unique minimizer exists, if there is some $\nu \in \mathbb{R}^Z$, $\nu \ll \mu$ such that $f(\nu) \neq \pm\infty$. Throughout this article we shall always assume that this is the case.

For Sect. 4 we require the following Lemma.

LEMMA 3 (Softmax and Softmin). For a parameter $\varepsilon > 0$ and $a \in \mathbb{R}^Z$ let

$$\text{softmax}(a, \varepsilon) := \varepsilon \log \left(\sum_{z \in Z} \exp(a(z)/\varepsilon) \right), \quad \text{softmin}(a, \varepsilon) := -\varepsilon \log \left(\sum_{z \in Z} \exp(-a(z)/\varepsilon) \right).$$

For $\varepsilon, \lambda > 0$ and $a, b \in \mathbb{R}^Z$ one has the relations

$$(1.3a) \quad \max(a) \leq \text{softmax}(a, \varepsilon) \leq \max(a) + \varepsilon \log |Z|,$$

$$(1.3b) \quad \min(a) - \varepsilon \log |Z| \leq \text{softmin}(a, \varepsilon) \leq \min(a),$$

$$(1.3c) \quad \min(a - b) - \lambda \log |Z| \leq \text{softmax}(a, \varepsilon) - \text{softmax}(b, \lambda) \leq \max(a - b) + \varepsilon \log |Z|,$$

$$(1.3d) \quad \min(a - b) - \varepsilon \log |Z| \leq \text{softmin}(a, \varepsilon) - \text{softmin}(b, \lambda) \leq \max(a - b) + \lambda \log |Z|.$$

Proof. The first line follows immediately from $0 \leq \exp(a(z)/\varepsilon) \leq \exp(\max a/\varepsilon)$. Line three then follows from $\min(a - b) \leq \max(a) - \max(b) \leq \max(a - b)$. The second and fourth line are implied by $\text{softmin}(a, \varepsilon) = -\text{softmax}(-a, \varepsilon)$. \square

2. Entropy Regularized Transport-Type Problems and Diagonal Scaling Algorithms.

2.1. Transport-Type Problems. For two probability measures $\mu \in \mathcal{P}(X)$ and $\nu \in \mathcal{P}(Y)$ the set $\Pi(\mu, \nu) := \{\pi \in \mathcal{P}(X \times Y) : P_X \pi = \mu, P_Y \pi = \nu\}$ is called the couplings or transport plans between μ and ν . A coupling π describes a rearrangement of the mass of μ into ν , $\pi(x, y)$ can be interpreted as the mass taken from x to y . Let $c \in \overline{\mathbb{R}}^{X \times Y}$ be a cost function, such that the cost of taking one unit of mass from $x \in X$ to $y \in Y$ is given by $c(x, y)$. The cost inflicted by a coupling π is then given by $\langle c, \pi \rangle$ and

the optimal transport problem between μ and ν is given by $\min\{\langle c, \pi \rangle \mid \pi \in \Pi(\mu, \nu)\}$. This means, we are looking for the most cost-efficient mass rearrangement between μ and ν . Note that for $\pi \in \mathbb{R}^{X \times Y}$ one can write $\iota_{\Pi(\mu, \nu)}(\pi) = \iota_{\{\mu\}}(\mathbb{P}_X \pi) + \iota_{\{\nu\}}(\mathbb{P}_Y \pi) + \iota_+(\pi)$ where the first two terms represent the marginal constraints and the last term ensures that π is non-negative. Then we can reformulate the problem as

$$(2.1) \quad \min_{\pi \in \mathbb{R}^{X \times Y}} \iota_{\{\mu\}}(\mathbb{P}_X \pi) + \iota_{\{\nu\}}(\mathbb{P}_Y \pi) + \langle c, \pi \rangle + \iota_+(\pi).$$

Recently it has been proposed to replace the constraints $\mathbb{P}_X \pi = \mu$ and $\mathbb{P}_Y \pi = \nu$ by soft constraints. This allows meaningful comparison between measures of different total mass. Such formulations were studied e.g. in [31] (see also [14] for more context). A particularly relevant choice for the soft constraints is the Kullback–Leibler divergence. A corresponding ‘unbalanced’ transport problem is given by

$$(2.2) \quad \min_{\pi \in \mathbb{R}^{X \times Y}} \lambda \cdot \text{KL}(\mathbb{P}_X \pi \mid \mu) + \lambda \cdot \text{KL}(\mathbb{P}_Y \pi \mid \nu) + \langle c, \pi \rangle + \iota_+(\pi).$$

where $\lambda > 0$ is a weighting parameter. Note that neither μ, ν nor π need to be probability measures in this case and each may have different total mass.

When $X = Y$ is a metric space with metric d , for $\lambda = 1$ and the cost function $c = d^2$, the square root of the optimal value of (2.2) yields the so called Gaussian Hellinger–Kantorovich (GHK) distance on \mathbb{R}_+^X , introduced in [31]. Similarly, for the cost function

$$(2.3) \quad c(x, y) := \begin{cases} -\log([\cos(d(x, y))]^2) & \text{if } d(x, y) < \pi/2 \\ +\infty & \text{else.} \end{cases}$$

one obtains the Wasserstein–Fisher–Rao (WFR) distance (or Hellinger–Kantorovich distance), introduced independently and simultaneously in [26, 13, 31]. WFR is the length distance induced by GHK [31].

Problems (2.1) and (2.2) share a common structure: in both we optimize over non-negative measures π on the product space $X \times Y$, there is a linear cost term $\langle c, \pi \rangle$ and two functions act on the marginals of π . They are prototypical examples of a family of transport-type optimization problems with a common functional structure that was introduced in [14]. The general structure is given in the following definition.

DEFINITION 4 (Generic Transport-Type Problem). *For two convex marginal functions $F_X : \mathbb{R}^X \rightarrow \overline{\mathbb{R}}$, $F_Y : \mathbb{R}^Y \rightarrow \overline{\mathbb{R}}$ and a cost function $c \in \overline{\mathbb{R}}^{X \times Y}$ the primal transport-type problem is given by:*

$$(2.4a) \quad \min_{\pi \in \mathbb{R}^{X \times Y}} E(\pi) \quad \text{with} \quad E(\pi) := F_X(\mathbb{P}_X \pi) + F_Y(\mathbb{P}_Y \pi) + \langle c, \pi \rangle + \iota_+(\pi)$$

The corresponding dual problem is given by:

$$(2.4b) \quad \max_{(\alpha, \beta) \in (\mathbb{R}^X, \mathbb{R}^Y)} J(\alpha, \beta) \quad \text{with} \quad J(\alpha, \beta) := -F_X^*(-\alpha) - F_Y^*(-\beta) - \iota_{\leq c}(\mathbb{P}_X^\top \alpha + \mathbb{P}_Y^\top \beta)$$

The indicator function $\iota_{\leq c}(\mathbb{P}_X^\top \alpha + \mathbb{P}_Y^\top \beta)$ denotes the classical optimal transport dual constraint $\alpha(x) + \beta(y) \leq c(x, y)$ for all $(x, y) \in X \times Y$ (see Section 1.3).

This family also covers Wasserstein gradient flows and the structure can be extended to multiple couplings to describe barycenter and multi-marginal problems (see [6, 14] for details). As indicated, the standard optimal transport problem (2.1) is obtained as a special case.

DEFINITION 5 (Standard Optimal Transport). *Problem (2.1) is a special case of Def. 4 with $F_X := \iota_{\{\mu\}}$ and $F_Y := \iota_{\{\nu\}}$. The primal and dual functional are given by:*

$$(2.5a) \quad E(\pi) = \iota_{\{\mu\}}(\mathbb{P}_X \pi) + \iota_{\{\nu\}}(\mathbb{P}_Y \pi) + \langle c, \pi \rangle + \iota_+(\pi)$$

$$(2.5b) \quad J(\alpha, \beta) = \langle \alpha, \mu \rangle + \langle \beta, \nu \rangle - \iota_{\leq c}(\mathbb{P}_X^\top \alpha + \mathbb{P}_Y^\top \beta)$$

Likewise, we can proceed for the unbalanced transport problem (2.2).

DEFINITION 6 (Unbalanced Optimal Transport with KL Fidelity). *Problem (2.2) is a special case of Def. 4 with $F_X := \lambda \cdot \text{KL}(\cdot \mid \mu)$ and $F_Y := \lambda \cdot \text{KL}(\cdot \mid \nu)$. The primal and dual functional are given by:*

$$(2.6) \quad E(\pi) = \lambda \cdot \text{KL}(\mathbb{P}_X \pi \mid \mu) + \lambda \cdot \text{KL}(\mathbb{P}_Y \pi \mid \nu) + \langle c, \pi \rangle + \iota_+(\pi)$$

$$(2.7) \quad J(\alpha, \beta) = -\lambda \cdot \text{KL}^*(-\alpha/\lambda) - \lambda \cdot \text{KL}^*(-\beta/\lambda) - \iota_{\leq c}(\mathbb{P}_X^\top \alpha + \mathbb{P}_Y^\top \beta)$$

2.2. Entropy Regularization and Diagonal Scaling Algorithms. Now we apply entropy regularization to the above transport-type problems (see Sect. 1.1 for references) and replace the non-negativity constraint in (2.4a) by the Kullback–Leibler divergence. For this we need to select some reference measure $\rho \in \mathbb{R}_+^{X \times Y}$. We then replace the term $\iota_+(\pi)$ in (2.4a) by $\varepsilon \cdot \text{KL}(\pi|\rho)$, where $\varepsilon > 0$ is a regularization parameter. Then one typically ‘pulls’ the linear cost term into the KL divergence:

$$(2.8) \quad \begin{aligned} \langle c, \pi \rangle + \varepsilon \text{KL}(\pi|\rho) &= \varepsilon \text{KL}(\pi|K) + \varepsilon \cdot (\rho(X \times Y) - K(X \times Y)) \\ \text{where } K \in \mathbb{R}_+^{X \times Y} &\text{ with } K(x, y) := \exp(-c(x, y)/\varepsilon) \cdot \rho(x, y). \end{aligned}$$

with the convention $\exp(-\infty) = 0$. K is called the kernel associated with c and the regularization parameter ε . For convenience we formally introduce the function

$$(2.9) \quad \text{get}K : \mathbb{R}_{++} \rightarrow \mathbb{R}^{X \times Y}, \quad \varepsilon \mapsto \exp(-c/\varepsilon) \odot \rho.$$

We obtain the regularized equivalent to Def. 4.

DEFINITION 7 (Regularized Generic Formulation).

$$(2.10a) \quad \min_{\pi \in \mathbb{R}_+^{X \times Y}} E(\pi) \quad \text{with} \quad E(\pi) := F_X(\text{P}_X \pi) + F_Y(\text{P}_Y \pi) + \varepsilon \text{KL}(\pi|K)$$

$$(2.10b) \quad \max_{(\alpha, \beta) \in (\mathbb{R}^X, \mathbb{R}^Y)} J(\alpha, \beta) \quad \text{with} \quad J(\alpha, \beta) := -F_X^*(-\alpha) - F_Y^*(-\beta) - \varepsilon \text{KL}^* \left([\text{P}_X^\top \alpha + \text{P}_Y^\top \beta] / \varepsilon | K \right)$$

Primal optimizers π^\dagger have the form

$$(2.11) \quad \pi^\dagger = \text{diag}(\exp(\alpha^\dagger/\varepsilon)) K \text{diag}(\exp(\beta^\dagger/\varepsilon))$$

where $(\alpha^\dagger, \beta^\dagger)$ are dual optimizers. Conversely, for dual optimizers $(\alpha^\dagger, \beta^\dagger)$, π^\dagger constructed as above is primal optimal [14].

Intuitively we see the relation between (2.4) and (2.10) as $\varepsilon \rightarrow 0$. For example, the term $\varepsilon \text{KL}^*([\text{P}_X^\top \alpha + \text{P}_Y^\top \beta] / \varepsilon | K)$ in (2.10b) can be interpreted as a smooth barrier function for the dual constraint $\text{P}_X^\top \alpha + \text{P}_Y^\top \beta \leq c$ in (2.4b). We refer to Sect. 1.1 for references to rigorous convergence results.

Under suitable assumptions problem (2.10b) can be solved by alternating optimization in α and β (see [14] for details). For fixed β , consider the KL^* -term:

$$\text{KL}_{X \times Y}^*([\text{P}_X^\top \alpha + \text{P}_Y^\top \beta] / \varepsilon | K) = \text{KL}_X^*(\alpha / \varepsilon | K \exp(\beta / \varepsilon)) + \sum_{(x, y) \in X \times Y} K(x, y) (\exp(\beta(y) / \varepsilon) - 1).$$

Note that the last term is constant w.r.t. α . Therefore, optimizing (2.10b) over α , for fixed β corresponds to maximizing

$$(2.12) \quad J_X(\alpha) = -F_X^*(-\alpha) - \varepsilon \text{KL}_X^*(\alpha / \varepsilon | K \exp(\beta / \varepsilon)),$$

where $K \exp(\beta / \varepsilon)$ denotes standard matrix vector multiplication. The corresponding primal problem consists of minimizing

$$(2.13) \quad E_X(\sigma) = F_X(\sigma) + \varepsilon \text{KL}_X(\sigma | K \exp(\beta / \varepsilon)).$$

This is a proximal step of F_X for the KL divergence with step size $1/\varepsilon$ (see Def. 2). So, by using the PD-optimality conditions between (2.12) and (2.13) (see e.g. [4, Thm. 19.1]), for a given β the primal optimizer σ^\dagger of (2.13) and the dual optimizer α^\dagger of (2.12) are given by

$$(2.14) \quad \sigma^\dagger = \text{prox}_\varepsilon F_X(K \exp(\beta / \varepsilon)), \quad \alpha^\dagger = \varepsilon \log(\sigma^\dagger \odot (K \exp(\beta / \varepsilon))),$$

Analogously, optimization w.r.t. β for fixed α is related to KL proximal steps of F_Y . Starting from some initial $\beta^{(0)}$, we can iterate alternating optimization to obtain a sequence $\beta^{(0)}, \alpha^{(1)}, \beta^{(1)}, \alpha^{(2)}, \dots$ as follows:

$$(2.15a) \quad \alpha^{(\ell+1)} := \varepsilon \log \left(\text{prox}_\varepsilon F_X(K \exp(\beta^{(\ell)} / \varepsilon)) \odot [K \exp(\beta^{(\ell)} / \varepsilon)] \right),$$

$$(2.15b) \quad \beta^{(\ell+1)} := \varepsilon \log \left(\text{prox}_\varepsilon F_Y(K^\top \exp(\alpha^{(\ell+1)} / \varepsilon)) \odot [K^\top \exp(\alpha^{(\ell+1)} / \varepsilon)] \right).$$

The algorithm becomes somewhat simpler when it is formulated in terms of the effective variables

$$(2.16) \quad u := \exp(\alpha/\varepsilon), \quad v := \exp(\beta/\varepsilon).$$

For more convenient notation we introduce the proxdiv operator of a function F and step size $1/\varepsilon$:

$$(2.17) \quad \text{proxdiv}_\varepsilon F : \sigma \mapsto \text{prox}_\varepsilon F(\sigma) \oslash \sigma$$

The iterations then become:

$$(2.18) \quad u^{(\ell+1)} := \text{proxdiv}_\varepsilon F_X(K v^{(\ell)}), \quad v^{(\ell+1)} := \text{proxdiv}_\varepsilon F_Y(K^\top u^{(\ell+1)}).$$

The primal-dual relation (2.11) then becomes $\pi^\dagger = \text{diag}(u^\dagger) K \text{diag}(v^\dagger)$, which is why u and v are often referred to as diagonal scaling factors.

REMARK 1. *Throughout this article, we will refer to the arguments of the dual functionals (2.4b) and (2.10b) as dual variables and denote them with (α, β) . The effective, exponentiated variables, introduced in (2.16), will be denoted by (u, v) and referred to as scaling factors.*

For future reference let us state the full scaling algorithm.

ALGORITHM 1 (Scaling Algorithm).

- 1: **function** SCALINGALGORITHM($\varepsilon, v^{(0)}$)
- 2: $K \leftarrow \text{get}K(\varepsilon)$; $v \leftarrow v^{(0)}$ // compute kernel, see (2.9); initialize scaling variable
- 3: **repeat**
- 4: $u \leftarrow \text{proxdiv}_\varepsilon F_X(K v)$; $v \leftarrow \text{proxdiv}_\varepsilon F_Y(K^\top u)$
- 5: **until** stopping criterion
- 6: **return** (u, v)
- 7: **end function**

The stopping criterion is typically a bound on the primal-dual gap between dual iterates $(\alpha, \beta) = \varepsilon \log(u, v)$ and primal iterate $\pi = \text{diag}(u) K \text{diag}(v)$, an error bound on the marginals of π (for standard optimal transport) or a pre-determined number of iterations.

With alternating iterations (2.15) or (2.18) a large family of functionals of form (2.10a) can be optimized, as long as the KL proximal steps of F_X and F_Y can be computed efficiently. A particularly relevant sub-family is, where F_X and F_Y are separable and are a sum of pointwise functions. Then the KL steps decompose into pointwise one-dimensional KL steps, see [14, Section 3.4] for details.

Since Section 4 focusses on the special case of entropy regularized optimal transport, let us explicitly state the corresponding functional and iterations.

DEFINITION 8 (Entropic Optimal Transport). *For marginals $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ and a cost function $c \in \mathbb{R}^{X \times Y}$ the entropy regularized optimal transport problem is obtained from Def. 7 by setting $F_X := \iota_{\{\mu\}}$, $F_Y := \iota_{\{\nu\}}$ (see Definition 5 for the unregularized functional). We find:*

$$(2.19a) \quad E(\pi) = \iota_{\{\mu\}}(\text{P}_X \pi) + \iota_{\{\nu\}}(\text{P}_Y \pi) + \varepsilon \text{KL}(\pi|K)$$

$$(2.19b) \quad J(\alpha, \beta) = \langle \alpha, \mu \rangle + \langle \beta, \nu \rangle - \varepsilon \text{KL}^* \left([\text{P}_X^\top \alpha + \text{P}_Y^\top \beta] / \varepsilon | K \right)$$

The proximal steps of F_X and F_Y are trivial (if K has non-empty columns and rows) and we recover the famous Sinkhorn iterations:

$$(2.20a) \quad \text{proxdiv}_\varepsilon F_X(\sigma) = \mu \oslash \sigma, \quad \text{proxdiv}_\varepsilon F_Y(\sigma) = \nu \oslash \sigma,$$

$$(2.20b) \quad u^{(\ell+1)} = \mu \oslash (K v^{(\ell)}), \quad v^{(\ell+1)} = \nu \oslash (K^\top u^{(\ell+1)}).$$

3. Stabilized Sparse Multi-Scale Algorithm. Throughout this section we combine four adaptations to the Algorithm 1 to overcome the limitations of a naive implementation outlined in Section 1.1.

3.1. Log-Domain Stabilization. When running Algorithm 1 with small regularization parameter ε , entries in the kernel K , and the scaling factors u and v may become both very small and very large, leading to numerical difficulties. However, under suitable conditions (e.g. standard optimal transport, finite cost function) it can be shown that the optimal dual variables (α, β) remain finite and have a stable limit as $\varepsilon \rightarrow 0$ ([16], see also Remark 7). In [27, 43] and others it was proposed to formulate the Sinkhorn iterations directly in terms of the dual variables, instead of the scaling factors. For example, an update of α would be performed as follows:

$$(3.1a) \quad \psi^{(\ell+1)}(x, y) = -c(x, y) + \beta^{(\ell)}(y), \quad \tilde{\psi}^{(\ell+1)}(x, y) = \psi^{(\ell+1)}(x, y) - \max_{y' \in Y} \psi^{(\ell+1)}(x, y')$$

$$(3.1b) \quad \alpha^{(\ell+1)}(x) = \varepsilon \log \mu(x) - \varepsilon \log \left(\sum_{y \in Y} \exp(\tilde{\psi}^{(\ell+1)}(x, y)/\varepsilon) \cdot \rho(x, y) \right) - \max_{y \in Y} \psi^{(\ell+1)}(x, y)$$

Subtracting the maximum from $\psi^{(\ell+1)}$ avoids large arguments in the exponential function. While this resolves the issue of extreme scaling factors, it perturbs the simple matrix multiplication structure of the algorithm and requires many additional evaluations of \exp and \log in each iteration.

As an alternative, we employ the redundant parametrization of the iterations as proposed in [14]. The scaling factors (u, v) , (2.16), are written as

$$(3.2) \quad u = \tilde{u} \odot \exp(\hat{\alpha}/\varepsilon), \quad v = \tilde{v} \odot \exp(\hat{\beta}/\varepsilon).$$

Our goal is to formulate iterations (2.18) directly in terms of (\tilde{u}, \tilde{v}) , while keeping $(\hat{\alpha}, \hat{\beta})$ unchanged during most iterations. The role of $(\hat{\alpha}, \hat{\beta})$ is to occasionally ‘absorb’ the large values of (u, v) such that (\tilde{u}, \tilde{v}) remain bounded. This leads to two types of iterations: stabilized iterations, during which only (\tilde{u}, \tilde{v}) are changed, and absorption iterations, during which (\tilde{u}, \tilde{v}) are absorbed into $(\hat{\alpha}, \hat{\beta})$. In this way, we can combine the simplicity of the scaling algorithm in terms of the scaling factor formulation with the numerical stability of the iterations in the log-domain formulation (3.1).

Analogous to the function $\text{get}K$, (2.9), we define the *stabilized kernel* as

$$(3.3a) \quad \text{get}\mathcal{K} : \mathbb{R}^X \times \mathbb{R}^Y \times \mathbb{R}_{++} \rightarrow \mathbb{R}^{X \times Y}, \quad (\alpha, \beta, \varepsilon) \mapsto \text{diag}(\exp(\alpha/\varepsilon)) \text{get}K(\varepsilon) \text{diag}(\exp(\beta/\varepsilon)),$$

$$(3.3b) \quad [\text{get}\mathcal{K}(\alpha, \beta, \varepsilon)](x, y) = \exp\left(-\frac{1}{\varepsilon} [c(x, y) - \alpha(x) - \beta(y)]\right) \cdot \rho(x, y).$$

The second line, (3.3b), should be used for numerical evaluation such that extreme values in (α, β) and c can cancel before exponentiation. Moreover, we introduce a stabilized version of the proxdiv operator:

$$(3.4) \quad \text{proxdiv}_\varepsilon F : (\sigma, \gamma) \mapsto \text{prox}_\varepsilon F(\exp(-\gamma/\varepsilon) \sigma) \odot \sigma$$

Note that the regular version of the proxdiv operator, (2.17), is a special case of the stabilized variant with $\gamma = 0$. With $K = \text{get}K(\varepsilon)$ and $\mathcal{K} = \text{get}\mathcal{K}(\hat{\alpha}, \hat{\beta}, \varepsilon)$ we observe that

$$(3.5a) \quad \text{proxdiv}_\varepsilon F(\mathcal{K} \tilde{v}, \hat{\alpha}) = \text{proxdiv}_\varepsilon F(K v) \odot \exp(\hat{\alpha}/\varepsilon),$$

$$(3.5b) \quad \text{proxdiv}_\varepsilon F(\mathcal{K}^\top \tilde{u}, \hat{\beta}) = \text{proxdiv}_\varepsilon F(K^\top u) \odot \exp(\hat{\beta}/\varepsilon).$$

For a threshold parameter $\tau > 0$ we formally state the stabilized variant of Algorithm 1.

ALGORITHM 2 (Stabilized Scaling Algorithm).

- 1: **function** SCALINGALGORITHMSTABILIZED($\varepsilon, \alpha^{(0)}, \beta^{(0)}$)
- 2: $(\hat{\alpha}, \hat{\beta}) \leftarrow (\alpha^{(0)}, \beta^{(0)}); (\tilde{u}, \tilde{v}) \leftarrow (1_X, 1_Y); \mathcal{K} \leftarrow \text{get}\mathcal{K}(\hat{\alpha}, \hat{\beta}, \varepsilon)$
- 3: **repeat**
- 4: **while** $[\|\tilde{u}\|_\infty \leq \tau] \wedge [\|\tilde{v}\|_\infty \leq \tau]$ **do**
- 5: $\tilde{u} \leftarrow \text{proxdiv}_\varepsilon F_X(\mathcal{K} \tilde{v}, \hat{\alpha}); \tilde{v} \leftarrow \text{proxdiv}_\varepsilon F_Y(\mathcal{K}^\top \tilde{u}, \hat{\beta})$ // stabilized iteration
- 6: **end while**
- 7: $(\hat{\alpha}, \hat{\beta}) \leftarrow (\hat{\alpha}, \hat{\beta}) + \varepsilon \cdot \log(\tilde{u}, \tilde{v}); (\tilde{u}, \tilde{v}) \leftarrow (1_X, 1_Y); \mathcal{K} \leftarrow \text{get}\mathcal{K}(\hat{\alpha}, \hat{\beta}, \varepsilon)$ // absorption iteration
- 8: **until** stopping criterion

```

9:    $(\hat{\alpha}, \hat{\beta}) \leftarrow (\hat{\alpha}, \hat{\beta}) + \varepsilon \cdot \log(\tilde{u}, \tilde{v})$ 
10:  return  $(\hat{\alpha}, \hat{\beta})$ 
11: end function

```

Any successive combination of stabilized iterations and absorption iterations in Algorithm 2 is mathematically equivalent to Algorithm 1, in the sense that they produce the same iterates (keep in mind (3.2–3.5)). But numerically, with finite floating point precision, combining both types of iterations can make a significant difference. In practice one can run several stabilized iterations in a row, occasionally checking whether (\tilde{u}, \tilde{v}) become too large or too small (see line 4), and perform an absorption iteration if required. This inflicts less computational overhead than the direct log-domain formulation (3.1) and largely preserves the simple matrix multiplication structure of the scaling algorithms.

In the definitions for the stabilized kernel, (3.3b), and proxdiv-operator, (3.4), there still appear exponentials of the form $\exp(\cdot/\varepsilon)$, which may explode as $\varepsilon \rightarrow 0$. Extending the max-argument trick in (3.1) to more general scaling algorithms entails similar questions. In the examples studied in Section 5 and those given in [14] we find however, that evaluation of the exponential $\exp(-\gamma/\varepsilon)$ can be avoided. For the special case of standard optimal transport ε no longer appears in the stabilized step.

3.2. ε -Scaling. It is empirically and theoretically well-known (cf. Section 1.1) that convergence of Algorithm 1 becomes slow as $\varepsilon \rightarrow 0$. A popular heuristic remedy is the so-called ε -scaling, where one subsequently solves the regularized problem with gradually decreasing values for ε . Let $\mathcal{E} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$ be a list of decreasing positive parameters. We extend Algorithm 2 as follows:

```

ALGORITHM 3 (Scaling Algorithm with  $\varepsilon$ -Scaling).
1: function SCALINGALGORITHM $\varepsilon$ SCALING( $\mathcal{E}, \alpha^{(0)}, \beta^{(0)}$ )
2:    $(\alpha, \beta) \leftarrow (\alpha^{(0)}, \beta^{(0)})$ 
3:   for  $\varepsilon \in \mathcal{E}$  do // iterate over list, from largest to smallest
4:      $(\alpha, \beta) \leftarrow$  SCALINGALGORITHMSTABILIZED( $\varepsilon, \alpha, \beta$ )
5:   end for
6:   return  $(\alpha, \beta)$ 
7: end function

```

The dual variable β is kept constant while changing ε , not the scaling factor v , because the optimal dual variables (α, β) usually have a stable limit as $\varepsilon \rightarrow 0$, while the scaling factors (u, v) diverge (see Sect. 1.1 and also Theorem 20).

So far, very little is known theoretically about the behaviour of ε -scaling for the Sinkhorn algorithm (cf. Section 1.1). Empirically, it is shown in Sect. 5.2 that ε -scaling is highly efficient and the number of required iterations does not increase exponentially. We observe that indeed it behaves similar as in the auction algorithm, as discussed in [27]. We work towards a theoretical quantification of this in Sect. 4.

Motivated by this, in practice we recommend a geometric decrease of ε and choose $\varepsilon_k = \varepsilon_0 \cdot \lambda^k$ such that ε_n is the desired final value, ε_0 is on the order of the maximal values in the cost function c and $\lambda \in (0, 1)$ is a geometric scaling factor, typically in $[0.5, 0.75]$. If λ is too small, iterations will start far from convergence after each change of ε , increasing the risk of numerical instabilities and requiring more iterations. On the other hand, if λ is too large, many stages of ε -scaling have to be performed, increasing numerical overhead.

3.3. Kernel Truncation. Storing the dense kernel K and computing dense matrix multiplications during the scaling iterations (2.18) requires a lot of memory and time on large problems. For several problems with particular structure, remedies have been proposed (Sect. 1.1). But these do not comprise non-standard cost functions, as the one used for the Wasserstein-Fisher-Rao distance, (2.3). Moreover they are not compatible with the log-stabilization (Section 3.1), thus a certain level of blur cannot be avoided. We are looking for a more flexible method to accelerate solving.

For many unregularized transport problems the optimal coupling π^\dagger is concentrated on a sparse subset of $X \times Y$. In fact, this is the underlying mechanism for the efficiency of most solvers discussed in Section 1.1. For the regularized problems the optimal coupling will usually be dense. This is due to the diverging derivative of the KL divergence at zero. However, as $\varepsilon \rightarrow 0$, the optimal coupling quickly converges to an unregularized solution (see Sect. 1.1, in particular [16, Thm. 5.8]). As $\varepsilon \rightarrow 0$, large parts

of the coupling will approach zero exponentially fast.

So while we will not be able to exactly solve the full problem, by solving suitable sparse sub-problems, we may still expect a reasonable approximation. We formalize the concept of a sparse sub-problem.

DEFINITION 9 (Sparse Sub-Problems). *Let F_X and F_Y be marginal functions and c be a cost function as in Definition 4 and let $\mathcal{N} \subset X \times Y$. We introduce:*

$$(3.6) \quad \hat{c}(x, y) := \begin{cases} c(x, y) & \text{if } (x, y) \in \mathcal{N}, \\ +\infty & \text{else.} \end{cases}, \quad \hat{K}(x, y) := \begin{cases} K(x, y) & \text{if } (x, y) \in \mathcal{N}, \\ 0 & \text{else.} \end{cases}$$

We call problems (2.4a) and (2.4b) with c replaced by \hat{c} the problems restricted to \mathcal{N} . This corresponds to adding the constraint $\text{spt } \pi \subset \mathcal{N}$ to the primal problem, and only enforcing the constraint $\alpha(x) + \beta(y) \leq c(x, y)$ on $(x, y) \in \mathcal{N}$ in the dual problem. The entropy regularized variants of the restricted problems are obtained through replacing K by \hat{K} in (2.10a) and (2.10b).

Clearly, when \mathcal{N} is sparse, then so is \hat{K} and the restricted regularized problem can be solved faster and with less memory. We now quantify the error inflicted by restriction.

PROPOSITION 10 (Restricted Kernel and Duality Gap). *Let $\varepsilon > 0$ and $\mathcal{N} \subset X \times Y$. Let E and J be unrestricted regularized primal and dual functionals with kernel K , as given in Definition 7, and let \hat{E} and \hat{J} be the functionals of the problems restricted to \mathcal{N} , with sparse kernel \hat{K} (see Def. 9).*

Further, let (α, β) be a pair of dual variables, let $u = \exp(\alpha/\varepsilon)$, $v = \exp(\beta/\varepsilon)$ be the corresponding scaling factors and let $\pi = \text{diag}(u) \hat{K} \text{diag}(v)$ be the corresponding (restricted) primal coupling.

Then we find for the primal-dual gap between π and (α, β) :

$$(3.7) \quad E(\pi) - J(\alpha, \beta) = \hat{E}(\pi) - \hat{J}(\alpha, \beta) + \varepsilon \sum_{(x, y) \in (X \times Y) \setminus \mathcal{N}} u(x) K(x, y) v(y).$$

Proof. For the primal score we find:

$$\begin{aligned} E(\pi) &= F_X(\text{P}_X \pi) + F_Y(\text{P}_Y \pi) + \varepsilon \sum_{(x, y) \in X \times Y} \left[\pi(x, y) \log \left(\frac{\pi(x, y)}{K(x, y)} \right) - \pi(x, y) + K(x, y) \right] \\ &= \hat{E}(\pi) + \varepsilon \sum_{(x, y) \in (X \times Y) \setminus \mathcal{N}} \underbrace{\left[\pi(x, y) \log \left(\frac{\pi(x, y)}{K(x, y)} \right) - \pi(x, y) + K(x, y) \right]}_{=0} \end{aligned}$$

Analogously, for the dual score we get:

$$\begin{aligned} J(\alpha, \beta) &= -F_X^*(-\alpha) - F_Y^*(-\beta) - \varepsilon \sum_{(x, y) \in X \times Y} K(x, y) \cdot (\exp([\alpha(x) + \beta(y)]/\varepsilon) - 1) \\ &= \hat{J}(\alpha, \beta) - \varepsilon \sum_{(x, y) \in (X \times Y) \setminus \mathcal{N}} K(x, y) \cdot \underbrace{\left(\exp([\alpha(x) + \beta(y)]/\varepsilon) - 1 \right)}_{=u(x)v(y)} \end{aligned}$$

Together we obtain $E(\pi) - J(\alpha, \beta) = \hat{E}(\pi) - \hat{J}(\alpha, \beta) + \varepsilon \sum_{(x, y) \in (X \times Y) \setminus \mathcal{N}} u(x) K(x, y) v(y)$. \square

That is, the primal-dual gap for the original full functionals is equal to the gap for the truncated functionals plus the ‘mass’ that we have chopped off by truncating K to \hat{K} , when using the scaling factors u and v . If some \mathcal{N} were known, on which most mass of the optimal π^\dagger is concentrated, it would be sufficient to solve the problem restricted to \mathcal{N} , to get a good approximate solution. The remaining challenge is, how to identify \mathcal{N} without knowing π^\dagger before.

We propose an iterative re-estimation of \mathcal{N} , based on current dual iterates and to combine this with the log-stabilized iteration scheme (Section 3.1) and the computation of the stabilized kernel, (3.3b). For a threshold parameter $\theta > 0$ we define the following functions:

$$(3.8) \quad \text{get}\mathcal{N}(\alpha, \beta, \varepsilon, \theta) := \{(x, y) \in X \times Y : \exp(-\frac{1}{\varepsilon}[c(x, y) - \alpha(x) - \beta(y)]) \geq \theta\}$$

$$(3.9) \quad [\text{get}\hat{K}(\alpha, \beta, \varepsilon, \theta)](x, y) := \begin{cases} \exp(-\frac{1}{\varepsilon}[c(x, y) - \alpha(x) - \beta(y)]) \rho(x, y) & \text{if } (x, y) \in \text{get}\mathcal{N}(\alpha, \beta, \varepsilon, \theta), \\ 0 & \text{else.} \end{cases}$$

$\text{get}\hat{K}$ can be used instead of $\text{get}\mathcal{K}$ in Algorithm 2. We refer to this as *absorption iteration with truncation*. For this combination one finds a simple bound for the primal-dual gap comparison of Proposition 10.

PROPOSITION 11 (Simple Duality Gap Estimate for Absorption Iterations with Truncation). *For a regularized problem as in Definition 7 with functionals E and J , let (u, v) be a pair of diagonal scaling factors and $(\alpha, \beta) = \varepsilon \cdot \log(u, v)$, let $(\hat{\alpha}, \hat{\beta})$ a pair of dual variables and (\tilde{u}, \tilde{v}) a pair of relative scaling factors such that $u = \tilde{u} \cdot \exp(\hat{\alpha}/\varepsilon)$ and $v = \tilde{v} \cdot \exp(\hat{\beta}/\varepsilon)$.*

Let further $\mathcal{N} = \text{get}\mathcal{N}(\hat{\alpha}, \hat{\beta}, \varepsilon, \theta)$, $\mathcal{K} = \text{get}\hat{K}(\hat{\alpha}, \hat{\beta}, \varepsilon, \theta)$, let \hat{E} and \hat{J} be the functionals restricted to \mathcal{N} (see Def. 9) and $\pi = \text{diag}(\tilde{u})\mathcal{K}\text{diag}(\tilde{v})$. Then $E(\pi) - J(\alpha, \beta) \leq \hat{E}(\pi) - \hat{J}(\alpha, \beta) + \|\tilde{u}\|_\infty \cdot \|\tilde{v}\|_\infty \cdot \theta \cdot \rho(X \times Y)$.

Proof. By virtue of Proposition 10

$$E(\pi) - J(\alpha, \beta) = \hat{E}(\pi) - \hat{J}(\alpha, \beta) + \sum_{(x,y) \in (X \times Y) \setminus \mathcal{N}} u(x) K(x, y) v(y).$$

For $(x, y) \in (X \times Y) \setminus \mathcal{N}$ one has $\exp(-\frac{1}{\varepsilon}[c(x, y) - \hat{\alpha}(x) - \hat{\beta}(y)]) < \theta$ and therefore

$$\begin{aligned} u(x) K(x, y) v(y) &= \tilde{u}(x) \exp\left(-\frac{1}{\varepsilon}[c(x, y) - \hat{\alpha}(x) - \hat{\beta}(y)]\right) \cdot \rho(x, y) \cdot \tilde{v}(y) \\ &\leq \tilde{u}(x) \tilde{v}(y) \theta \rho(x, y). \end{aligned}$$

The result follows by bounding $\tilde{u}(x) \leq \|\tilde{u}\|_\infty$, $\tilde{v}(y) \leq \|\tilde{v}\|_\infty$ and summing over $(X \times Y) \setminus \mathcal{N}$. \square

This implies that in Algorithm 2 with truncation the additional duality gap error due to the sparse kernel is bounded by $\|\tilde{u}^{(\ell)}\|_\infty \cdot \|\tilde{v}^{(\ell)}\|_\infty \cdot \theta \cdot \rho(X \times Y)$. In particular, before every stabilized iteration the error is bounded by $\tau^2 \cdot \theta \cdot \rho(X \times Y)$ and after every absorption iteration it is bounded by $\theta \cdot \rho(X \times Y)$. This bound is easy to evaluate and does not require to sum over $(X \times Y) \setminus \mathcal{N}$, as the exact expression in Proposition 10. We find that in practice this truncation error bound can be kept much smaller than the remaining primal-dual gap $\hat{E}(\pi) - \hat{J}(\alpha, \beta)$.

In general the stabilized iteration scheme *with truncation* might not converge. However, by Proposition 11, if one regularly performs an absorption iteration before $\|\tilde{u}^{(\ell)}\|_\infty \cdot \|\tilde{v}^{(\ell)}\|_\infty$ becomes too large, the potential oscillations in the primal iterates and primal and dual functionals are numerically negligible.

3.4. Multi-Scale Scheme. Finally, we propose to combine the stabilized sparse iterations with a hierarchical multi-scale scheme, analogous to the ideas in [34, 41, 35].

This serves two purposes: First, a hierarchical representation of the problem allows to determine the truncated sparse stabilized kernel $\text{get}\hat{K}$, (3.9), with a coarse-to-fine tree search, without explicitly testing all pairs $(x, y) \in X \times Y$. The second reason is to make the combination of ε -scaling (Algorithm 3) with the truncated stabilized scheme more efficient. For a fixed threshold θ , while ε is large, the support of the truncated kernel $\text{get}\hat{K}$ will contain many variables. At the same time, due to the blur induced by the regularization, the primal iterates will not provide a sharply resolved assignment. Solving the problems with large ε -value on a coarser grid reduces the number of required variables, without losing much spatial accuracy. As ε decreases, so does the number of variables in $\text{get}\hat{K}$ (since the exponential function decreases faster), and the resolution of X and Y can be increased. Therefore, it is reasonable to coordinate the reduction of ε with increasing the spatial resolution of the transport problem, until the desired regularization and resolution are attained.

We will now briefly recall the hierarchical representation of a transport problem from [41].

DEFINITION 12 (Hierarchical Partition and Multi-Scale Measure Approximation [41]). *For a discrete set X a hierarchical partition is an ordered tuple $(\mathcal{X}_0, \dots, \mathcal{X}_I)$ of partitions of X where $\mathcal{X}_0 = \{\{x\} : x \in X\}$ is the trivial partition of X into singletons and each subsequent level is generated by merging cells from the previous level, i.e. for $i \in \{1, \dots, I\}$ and any $\mathbf{x} \in \mathcal{X}_i$ there exists some $\hat{\mathcal{X}} \subset \mathcal{X}_{i-1}$ such that $\mathbf{x} = \bigcup_{\hat{\mathbf{x}} \in \hat{\mathcal{X}}} \hat{\mathbf{x}}$. For simplicity we assume that the coarsest level is the trivial partition into one set: $\mathcal{X}_I = \{X\}$. We call $I > 0$ the depth of \mathcal{X} .*

This implies a directed tree graph with vertex set $\bigcup_{i=0}^I \mathcal{X}_i$. For $i, j \in \{0, \dots, I\}$, $i < j$ we say $\mathbf{x} \in \mathcal{X}_i$ is a descendant of $\mathbf{x}' \in \mathcal{X}_j$ when $\mathbf{x} \subset \mathbf{x}'$. We call \mathbf{x} a child of \mathbf{x}' for $i = j - 1$, and a leaf for $i = 0$. For some $\mu \in \mathbb{R}^X$ its multi-scale measure approximation is the tuple (μ_0, \dots, μ_I) of measures $\mu_i \in \mathbb{R}^{\mathcal{X}_i}$

defined by $\mu_i(\hat{X}) = \mu(\bigcup_{\mathbf{x} \in \hat{X}} \mathbf{x})$ for all subsets $\hat{X} \subset \mathcal{X}_i$ and $i = 0, \dots, I$. For convenience we often identify X with the finest partition level \mathcal{X}_0 and μ with μ_0 .

DEFINITION 13 (Hierarchical Dual Variables and Costs [41]). *Let X and Y be discrete sets with hierarchical partitions $\mathcal{X} = (\mathcal{X}_0, \dots, \mathcal{X}_I)$, $\mathcal{Y} = (\mathcal{Y}_0, \dots, \mathcal{Y}_I)$ of depth I , let $\alpha \in \mathbb{R}^X$ and $\beta \in \mathbb{R}^Y$ be functions over X and Y , and let $c \in \mathbb{R}^{X \times Y}$ be a cost function.*

Then we define the extension $\hat{\alpha} = (\hat{\alpha}_0, \dots, \hat{\alpha}_I)$ of α onto the full partition \mathcal{X} by

$$(3.10) \quad \hat{\alpha}_i(\mathbf{x}) = \max_{x \in \mathbf{x}} \alpha(x) = \begin{cases} \alpha(x) & \text{if } i = 0 \text{ and } \mathbf{x} = \{x\} \text{ for some } x \in X, \\ \max_{\mathbf{x}' \in \text{children}(\mathbf{x})} \hat{\alpha}_{i-1}(\mathbf{x}') & \text{if } i > 0, \end{cases}$$

for $i \in \{0, \dots, I\}$ and $\mathbf{x} \in \mathcal{X}_i$ and analogous for $\hat{\beta}$ and β . Similarly, define an extension \hat{c} of c by

$$(3.11) \quad \hat{c}_i(\mathbf{x}, \mathbf{y}) = \min_{(x,y) \in \mathbf{x} \times \mathbf{y}} c(x, y)$$

for $i \in \{0, \dots, I\}$, $\mathbf{x} \in \mathcal{X}_i$ and $\mathbf{y} \in \mathcal{Y}_i$.

For $i \in \{0, \dots, I\}$, $x \in \mathbf{x} \in \mathcal{X}_i$, $y \in \mathbf{y} \in \mathcal{Y}_i$ we find

$$(3.12) \quad \hat{c}_i(\mathbf{x}, \mathbf{y}) - \hat{\alpha}_i(\mathbf{x}) - \hat{\beta}_i(\mathbf{y}) \leq c(x, y) - \alpha(x) - \beta(y).$$

Now we can implement a hierarchical tree-search for $\text{get}\mathcal{N}$ (and analogously $\text{get}\hat{K}$).

ALGORITHM 4 (Hierarchical Search for $\text{get}\mathcal{N}$).

```

1: function  $\text{get}\mathcal{N}(\alpha, \beta, \varepsilon, \theta)$ 
2:    $(\hat{\alpha}, \hat{\beta}) \leftarrow$  hierarchical extensions of  $(\alpha, \beta)$  // see (3.10)
3:    $\mathcal{N} \leftarrow \text{SCALLCELL}(\hat{\alpha}, \hat{\beta}, \varepsilon, \theta, I, \{X\}, \{Y\})$  // call on coarsest partition level
4: end function

5: function  $\text{SCANCELL}(\hat{\alpha}, \hat{\beta}, \varepsilon, \theta, i, \mathbf{x}, \mathbf{y})$ 
6:    $\mathcal{N}' \leftarrow \emptyset$  // temporary variable for result
7:   if  $\hat{c}_i(\mathbf{x}, \mathbf{y}) - \hat{\alpha}_i(\mathbf{x}) - \hat{\beta}_i(\mathbf{y}) \leq -\varepsilon \cdot \log \theta$  then // if cell cannot be ruled out at this level
8:     if  $i > 0$  then // if not yet at finest level, check on all children
9:       for  $(\mathbf{x}', \mathbf{y}') \in \text{children}(\mathbf{x}) \times \text{children}(\mathbf{y})$  do
10:         $\mathcal{N}' \leftarrow \mathcal{N}' \cup \text{SCANCELL}(\hat{\alpha}, \hat{\beta}, \varepsilon, \theta, i - 1, \mathbf{x}', \mathbf{y}')$ 
11:      end for
12:     else // if at finest level, add variable
13:        $\mathcal{N}' \leftarrow \mathcal{N}' \cup (\mathbf{x} \times \mathbf{y})$  // recall  $\mathbf{x} = \{x\}$ ,  $\mathbf{y} = \{y\}$  for some  $(x, y) \in X \times Y$  at  $i = 0$ 
14:     end if
15:   end if
16:   return  $\mathcal{N}'$ 
17: end function

```

From (3.12) follows directly that Algorithm 4 implements (3.8).

In many applications the discrete sets X and Y are point clouds in \mathbb{R}^d and the hierarchical partitions are 2^d -trees over X and Y (see e.g. [40]). The cost function c is often originally defined on the whole product space $\mathbb{R}^d \times \mathbb{R}^d$ (such as the squared Euclidean distance). For the validity of Algorithm 4 it suffices if $\hat{c}_i(\mathbf{x}, \mathbf{y}) \leq \min_{(x,y) \in \mathbf{x} \times \mathbf{y}} c(x, y)$. This allows to avoid computing (and storing) the full cost matrix $c \in \mathbb{R}^{X \times Y}$ and the explicit minimizations in (3.11). c and lower bounds on \hat{c}_i can be computed on-demand directly using the tree-structure.

The second purpose of the multi-scale scheme is the combination with ε -scaling. As explained above, the purpose is to reduce the number of variables while ε is large. For an illustration see Fig. 1. For this, we divide the list \mathcal{E} of regularization parameters ε into multiple lists $(\mathcal{E}_0, \dots, \mathcal{E}_I)$, with the largest values in \mathcal{E}_I and the smallest (and final) values in \mathcal{E}_0 , and sorted from largest to smallest within each \mathcal{E}_i . Then, for every i from I down to 0 we perform ε -scaling with list \mathcal{E}_i at hierarchical level i , using the dual solution at each level as initialization at the next stage. The full algorithm, combining log-stabilization, ε -scaling, kernel truncation and the multi-scale scheme, is sketched next.

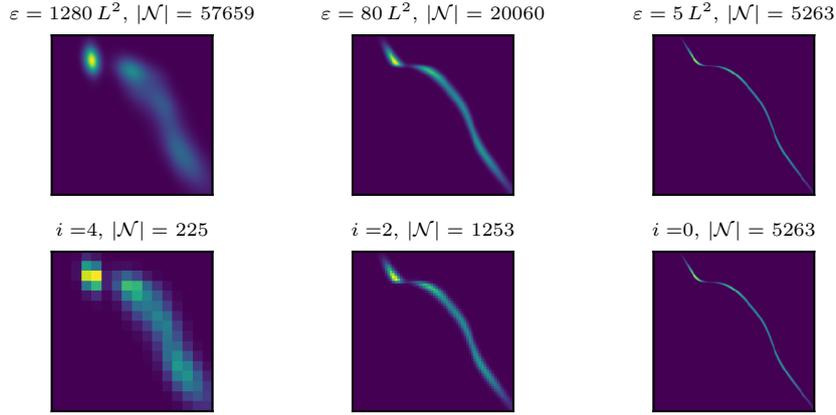


FIG. 1. ε -scaling, truncated kernels and multi-scale scheme. $X = Y$ is a uniform one-dimensional grid, representing $[0, 1]$, $|X| = 256$, $h = 256^{-1}$. μ and ν are smooth mixtures of Gaussians. **Top row** Density of optimal coupling π^\dagger on X^2 for various ε . $|\mathcal{N}|$ is the number of variables in the truncated, stabilized kernel for fixed $\theta = 10^{-10}$. As ε decreases, so does $|\mathcal{N}|$, since π^\dagger becomes more concentrated. **Bottom row** Optimal couplings for same ε as top row, but for different levels i of hierarchical partitions. i and ε were chosen to keep number of variables per $x \in X$ approximately constant. For high ε (and i) $|\mathcal{N}|$ is now dramatically lower. While π^\dagger is ‘pixelated’ for high i , due to blur, it provides roughly the same spatial information as the top row. Images in third column are identical.

ALGORITHM 5 (Full Algorithm).

```

1: function SCALINGALGORITHMFULL( $(\mathcal{E}_0, \dots, \mathcal{E}_I), \theta$ )
2:    $i = I$ ;  $(\alpha, \beta) \leftarrow ((0), (0))$  // initialize scale counter and dual variables
3:   while  $i \geq 0$  do
4:     // solve problem at scale  $i$  with  $\varepsilon$ -scaling over  $\mathcal{E}_i$ 
5:     for  $\varepsilon \in \mathcal{E}_i$  do // iterate over list, from largest to smallest
6:        $(\alpha, \beta) \leftarrow$  SCALINGALGORITHMSTABILIZED( $i, \varepsilon, \theta, \alpha, \beta$ )
7:     end for
8:      $i \leftarrow i - 1$ 
9:     if  $i \geq 0$  then // refine dual variables
10:       $(\alpha, \beta) \leftarrow$  REFINEDUALS( $i, \alpha, \beta$ )
11:    end if
12:  end while
13:  return  $(\alpha, \beta)$ 
14: end function

```

Note: SCALINGALGORITHMSTABILIZED refers to calling Algorithm 2 for solving the problem at scale i , with $\text{get}\mathcal{K}$ replaced by $\text{get}\hat{K}$, (3.9), with threshold θ , implemented according to Algorithm 4. Accordingly, two arguments i and θ were added. REFINEDUALS initializes the dual variables (α, β) at level i by setting the values at \mathbf{x} to the previous values at $\text{parent}(\mathbf{x})$ for all cells \mathbf{x} in \mathcal{X}_i .

REMARK 2 (Hierarchical Representation of F_X, F_Y). To solve the problem at hierarchical scale i , not only do we need a coarse version of c , as given in (3.11). In addition we need hierarchical versions of the marginal functions F_X, F_Y , see (2.10). An appropriate choice is often clear from the context of the problem. For example, for an optimal transport problem between μ and ν , see Def. 8, we set $F_{\mathcal{X}_i} = \iota_{\{\mu_i\}}$, where μ_i is taken from the multi-scale measure approximation of μ (see Def. 12). For the unbalanced transport problem with KL fidelity, Def. 6, we use $F_{\mathcal{X}_i} = \lambda \cdot \text{KL}_{\mathcal{X}_i}(\cdot | \mu_i)$.

This completes the modifications of the diagonal scaling algorithm. Their usefulness will be demonstrated numerically in Sect. 5.

4. Analogy between Sinkhorn and Auction Algorithm. In this section we develop a new complexity analysis of the Sinkhorn algorithm and examine the efficiency of ε -scaling. In [27] an intuitive similarity between the Sinkhorn algorithm for the entropy regularized linear assignment problem and the

auction algorithm was pointed out. This similarity motivates our approach.

In this section we only consider the standard Sinkhorn algorithm (as opposed to general scaling algorithms), since the auction algorithm solves the linear assignment problem and assumptions on fixed marginals μ, ν are required for our analysis.

The auction algorithm is briefly recalled in Section 4.1. In Section 4.2 we introduce an asymmetric variant of the Sinkhorn algorithm, that is more similar to the original auction algorithm and provide an analogous worst-case estimate for the number of iterations until a given precision is achieved. A stability result for the dual optimal solutions under change of the regularization parameter ε is given in Section 4.3 and we discuss how it relates to ε -scaling in Sect. 4.4.

4.1. Auction Algorithm. For the sake of self-containedness, in this section we briefly recall the auction algorithm and its basic properties. Note that compared to the original presentation (e.g. [9]) we flipped the overall sign for compatibility with the notion of optimal transport.

In the following we consider a linear assignment problem, i.e. an optimal transport problem between two discrete sets X, Y with equal cardinality $|X| = |Y| = N$ where the marginals $\mu \in \mathbb{R}_+^X, \nu \in \mathbb{R}_+^Y$ are the counting measures. For simplicity we assume that the cost function $c \in \mathbb{R}_+^{X \times Y}$ is finite and non-negative.

The main loop of the auction algorithm is divided into two parts: During the bidding phase, elements of X that are unassigned determine their locally most attractive counterpart in Y (taking into account the current dual variables) and submit a bid for them. During the assignment phase, all elements of Y that received at least one bid, pick the most attractive one and change the current assignment accordingly. A formal description is given in the following.

ALGORITHM 6 (Auction Algorithm).

```

1: function AUCTIONALGORITHM( $\beta^{(0)}$ )
2:    $\pi \leftarrow 0_{X \times Y}; \beta \leftarrow \beta^{(0)}$  // initialize variables: 'empty' primal coupling, zero dual variable
3:   while  $\pi(X \times Y) < N$  do
4:      $B(y) \leftarrow \emptyset$  for all  $y \in Y$  // start bidding phase: initialize empty bid lists
5:     for  $x \in \{x' \in X : \pi(\{x'\} \times Y) = 0\}$  do // iterate over unassigned  $x$ 
6:        $y \leftarrow \operatorname{argmin}_{y' \in Y} [c(x, y') - \beta(y')]$  // pick some element from argmin
7:        $\alpha(x) \leftarrow c(x, y) - \beta(y)$  // set dual variable
8:        $B(y) \leftarrow B(y) \cup \{x\}$  // submit bid to  $y$ , i.e. add  $x$  to bid list of  $y$ 
9:     end for
10:    for  $y \in \{y' \in Y : B(y') \neq \emptyset\}$  do // assignment phase: iterate over all  $y$  that received bids
11:       $\pi(\cdot, y) \leftarrow 0$  // set column of coupling to zero
12:       $x \leftarrow \operatorname{argmin}_{x' \in B(y)} [c(x', y) - \alpha(x')]$  // find best bidder, pick one if multiple
13:       $\beta(y) \leftarrow c(x, y) - \alpha(x) - \varepsilon; \pi(x, y) \leftarrow 1$  // update dual variable and coupling
14:    end for
15:  end while
16:  return  $(\pi, (\alpha, \beta))$ 
17: end function

```

REMARK 3. In the above algorithm, line 7 is usually replaced by $\alpha(x) \leftarrow \min_{y' \in Y \setminus \{y\}} [c(x, y') - \beta(y')]$, which in practice may reduce the number of iterations. It does not affect the following worst-case analysis however, therefore we keep the simpler version.

We briefly summarize the main properties of the algorithm.

PROPOSITION 14. With $\varepsilon > 0$ and $\beta^{(0)} = 0_Y$, Algorithm 6 has the following properties:

- (i) α is increasing, β is decreasing.
- (ii) After each assignment phase one finds $\alpha(x) + \beta(y) \leq c(x, y)$ and $[\pi(x, y) > 0] \Rightarrow [\alpha(x) + \beta(y) \geq c(x, y) - \varepsilon]$. The latter property is called ε -complementary slackness.
- (iii) The primal iterate satisfies $P_X \pi \leq \mu$ and $P_Y \pi \leq \nu$.
- (iv) The algorithm terminates after at most $N \cdot (C/\varepsilon + 1)$ iterations, where $C = \max c$.

For a proof see for example [9]. From ε -complementary slackness we deduce the following result.

COROLLARY 15. Upon convergence, the primal-dual gap of π and (α, β) , cf. Def. 5, is bounded by $\langle c, \pi \rangle - (\langle \mu, \alpha \rangle + \langle \nu, \beta \rangle) \leq N \cdot \varepsilon$. If c is integer and $\varepsilon < 1/N$, then the final primal coupling is optimal.

REMARK 4 (ε -Scaling for the Auction Algorithm). *During the auction algorithm it may happen that several elements in X compete for the same target $y \in Y$, leading to the minimal decrease of $\beta(y)$ by ε in each iteration. This phenomenon has been dubbed ‘price haggling’ [10] and can cause poor practical performance of the algorithm, close to the worst-case iteration bound. The impact of price haggling can be reduced by the ε -scaling technique, where the algorithm is successively run with a sequence of decreasing values for ε , each time using the final value of β as initialization of the next run (see also Algorithm 3). With this technique the factor C/ε in the iteration bound can essentially be reduced to a factor $\log(C/\varepsilon)$. An analysis of the ε -scaling technique for more general min-cost-flow problems can be found in [10].*

4.2. Asymmetric Sinkhorn Algorithm and Iteration Bound. We now introduce a slightly modified variant of the standard Sinkhorn algorithm, derive an iteration bound and make a comparison with the auction algorithm. We emphasize that this modification is primarily made to facilitate theoretical study of the algorithm and to understand why convergence becomes slow as $\varepsilon \rightarrow 0$. We do not advocate its merits in an actual implementation.

For $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ and a cost function $c \in \mathbb{R}_+^{X \times Y}$ we consider the entropic optimal transport problem (Def. 8). Set the reference measure ρ for regularization, see (2.8), to the product measure $\rho(x, y) = \mu(x) \cdot \nu(y)$. We state the modified Sinkhorn algorithm with parameter $q_{\text{target}} \in (0, 1)$ that measures how much mass has to be assigned.

ALGORITHM 7 (Asymmetric Sinkhorn Algorithm).

```

1: function ASYMMETRICSINKHORN( $\varepsilon, v^{(0)}, q_{\text{target}}$ )
2:    $K \leftarrow \text{getK}(\varepsilon); v = v^{(0)}$  // compute kernel, initialize scaling factor
3:   repeat
4:      $u \leftarrow \mu \otimes (K v); \hat{v} \leftarrow \nu \otimes (K^\top u)$ 
5:      $v \leftarrow \min\{v, \hat{v}\}$  // element-wise minimum
6:      $\pi \leftarrow \text{diag}(u) K \text{diag}(v); q \leftarrow \pi(X \times Y)$  // update coupling and ‘assigned mass’-fraction  $q$ 
7:   until  $q \geq q_{\text{target}}$ 
8:   return  $(\pi, (u, v))$ 
9: end function

```

The only differences to the standard Sinkhorn algorithm (given by Algorithm 1 with proxdiv-operators (2.20)) lie in line 5 and in the choice of the specific stopping criterion (see Remark 5 for a discussion). In the standard algorithm one would set $v \leftarrow \hat{v}$. The modification implies that v is monotonously decreasing, which implies the following result for Algorithm 7 in the spirit of Proposition 14. This monotonicity is crucial for bounding the number of iterations (see also Remark 6).

Throughout this section, for clarity, we enumerate the iterates u, v , as well as the auxiliary variables \hat{v}, π and q in Algorithm 7, starting with $v^{(0)}$ and proceeding with $(u^{(1)}, v^{(1)}, \hat{v}^{(1)}, \pi^{(1)}, q^{(1)}), \dots$, similar to formulas (2.18) and the corresponding dual variable iterates $(\alpha^{(\ell)}, \beta^{(\ell)}, \hat{\beta}^{(\ell)}) = \varepsilon \cdot \log(u^{(\ell)}, v^{(\ell)}, \hat{v}^{(\ell)})$.

PROPOSITION 16 (Monotonicity of Asymmetric Sinkhorn Algorithm).

- (i) u and $\alpha = \varepsilon \log u$ are increasing, v and $\beta = \varepsilon \log v$ are decreasing, q is increasing.
- (ii) $P_X \pi \leq \mu$ and $P_Y \pi \leq \nu$. We say π is sub-feasible.
- (iii) There exists some $y^* \in Y$ such that $v(y^*) = v^{(0)}(y^*)$ for all iterations.

Proof. By construction we have $v^{(\ell+1)} \leq v^{(\ell)}$. Consequently $K v^{(\ell+1)} \leq K v^{(\ell)}$ and thus $u^{(\ell+1)} \geq u^{(\ell)}$ and eventually $\hat{v}^{(\ell+1)} \leq \hat{v}^{(\ell)}$.

After updating $u^{(\ell+1)}$ the row constraints are satisfied. That is $P_X \text{diag}(u^{(\ell+1)}) K \text{diag}(v^{(\ell)}) = \mu$. Since $v^{(\ell+1)}$ is only decreased (i.e. if the corresponding column constraint is violated from above), afterwards the iterate $\pi^{(\ell+1)}$ is sub-feasible.

Since $\hat{v}^{(\ell)}$ is decreasing, it follows that if $v^{(\ell)}(y) = \hat{v}^{(\ell)}(y)$ for some $y \in Y$, then $v^{(k)}(y) = \hat{v}^{(k)}(y)$ for all $k \geq \ell$. Let $Y^{(\ell)} = \{y \in Y : v^{(\ell)}(y) = \hat{v}^{(\ell)}(y)\}$. Then $Y^{(\ell)} \subset Y^{(\ell+1)}$. Conversely, if $y \notin Y^{(\ell)}$, then $v^{(\ell)}(y) < \hat{v}^{(\ell)}(y)$ and therefore $v^{(\ell)}(y) = v^{(0)}(y)$.

Let now $q^{(\ell)}(y) := v^{(\ell)}(y) [K^\top u^{(\ell)}](y)$. If $y \in Y^{(\ell+1)}$, then $q^{(\ell+1)}(y) = \nu(y) \geq q^{(\ell)}(y)$ (as $q^{(\ell)}(y)$ can never exceed $\nu(y)$). If $y \notin Y^{(\ell+1)}$, then $v^{(\ell+1)}(y) = v^{(\ell)}(y) = v^{(0)}(y)$ and since $u^{(\ell)}$ is increasing, we find $q^{(\ell+1)}(y) \geq q^{(\ell)}(y)$. We obtain $q^{(\ell+1)} = \sum_{y \in Y} q^{(\ell+1)}(y) \geq q^{(\ell)}$.

When $Y^{(\ell)} \neq Y$, there exists some $y^* \in Y$ with $y^* \notin Y^{(k)}$, $v^{(k)}(y^*) = v^{(0)}(y^*)$ for $k \in \{1, \dots, \ell\}$. If

$Y^{(\ell)} = Y$, then $\hat{v}^{(\ell)} = v^{(\ell)} \leq v^{(\ell-1)}$. By construction one has $(u^{(\ell)})^\top K v^{(\ell-1)} = \mu(X)$ and $(u^{(\ell)})^\top K \hat{v}^{(\ell)} = \nu(Y) = \mu(X)$. So if $Y^{(\ell)} = Y$, in fact $v^{(\ell)} = v^{(\ell-1)}$. Consequently, there exists some $y^* \in Y$ with $v^{(\ell)}(y^*) = v^{(0)}(y^*)$ for all iterations. \square

Let us further investigate the increments of the dual variable iterates $\alpha^{(\ell)} = \varepsilon \log(u^{(\ell)})$.

LEMMA 17 (Minimal Increment of $\alpha^{(\ell)}$). *For $\ell \geq 1$ have $\langle \alpha^{(\ell+1)} - \alpha^{(\ell)}, \mu \rangle \geq \varepsilon(1 - q^{(\ell)})$.*

Proof. Recall that $\pi^{(\ell)} = \text{diag}(u^{(\ell)}) K \text{diag}(v^{(\ell)})$, and introduce $\pi'^{(\ell)} = \text{diag}(u^{(\ell+1)}) K \text{diag}(v^{(\ell)})$. Consider the following evaluations of the dual functional:

$$\begin{aligned} J(\alpha^{(\ell)}, \beta^{(\ell)}) &= \langle \alpha^{(\ell)}, \mu \rangle + \langle \beta^{(\ell)}, \nu \rangle - \varepsilon \cdot \pi^{(\ell)}(X \times Y) + \varepsilon \cdot K(X \times Y) \\ J(\alpha^{(\ell+1)}, \beta^{(\ell)}) &= \langle \alpha^{(\ell+1)}, \mu \rangle + \langle \beta^{(\ell)}, \nu \rangle - \varepsilon \cdot \pi'^{(\ell)}(X \times Y) + \varepsilon \cdot K(X \times Y) \end{aligned}$$

Note that $\pi^{(\ell)}(X \times Y) = q^{(\ell)}$, $\pi'^{(\ell)}(X \times Y) = 1$ and since going from $\alpha^{(\ell)}$ to $\alpha^{(\ell+1)}$ corresponds to a block-wise dual maximization have $J(\alpha^{(\ell+1)}, \beta^{(\ell)}) \geq J(\alpha^{(\ell)}, \beta^{(\ell)})$. The claim follows. \square

With these tools we can bound the total number of iterations to reach a given precision.

PROPOSITION 18 (Iteration Bound for the Asymmetric Sinkhorn Algorithm). *Initializing with $\beta^{(0)} = 0_Y \Leftrightarrow v^{(0)} = 1_Y$, for a given $q_{\text{target}} \in (0, 1)$ the number of iterations n necessary to achieve $q^{(n)} \geq q_{\text{target}}$ is bounded by $n \leq 2 + \frac{C}{\varepsilon(1-q_{\text{target}})}$ where $C = \max c$. Moreover, $\langle u^{(\ell)}, \mu \rangle \leq \exp(C/\varepsilon)$ for all iterates $\ell \geq 1$.*

Proof. Let us look at the first ‘bid’ $\alpha^{(1)}$. With $c \geq 0$ we have

$$(4.1) \quad \alpha^{(1)}(x) = \varepsilon \log \left(\frac{\mu(x)}{[K v^{(0)}](x)} \right) = \varepsilon \log \left(\frac{1}{\sum_{y \in Y} \nu(y) \exp(-c(x, y)/\varepsilon)} \right) \geq \varepsilon \log \left(\frac{1}{\sum_{y \in Y} \nu(y)} \right) = 0.$$

By virtue of Proposition 16 get $q^{(\ell)} \leq q^{(n)}$ for $\ell \leq n$. With Lemma 17 this implies $\langle \alpha^{(n)} - \alpha^{(1)}, \mu \rangle \geq \sum_{\ell=1}^{n-1} \varepsilon \cdot (1 - q^{(\ell)}) \geq \varepsilon \cdot (n-1) \cdot (1 - q^{(n)})$ and with (4.1)

$$(4.2) \quad \langle \alpha^{(n)}, \mu \rangle \geq \varepsilon \cdot (n-1) \cdot (1 - q^{(n)}).$$

From Proposition 16 we know that there is some $y^* \in Y$ with $v^{(\ell)}(y^*) = 1 \leq \hat{v}^{(\ell+1)}(y^*)$ for all iterates $\ell \geq 0$. So

$$1 \leq \hat{v}^{(\ell+1)}(y^*) = \frac{\nu(y^*)}{[K^\top u^{(\ell+1)}](y^*)} = \frac{1}{\sum_{x \in X} \exp\left(-\frac{1}{\varepsilon}[c(x, y^*) - \alpha^{(\ell+1)}(x)]\right) \mu(x)},$$

from which we infer $\exp(-C/\varepsilon) \cdot \langle \exp(\alpha^{(\ell+1)}/\varepsilon), \mu \rangle \leq 1$, i.e. $\langle u^{(\ell+1)}, \mu \rangle \leq \exp(C/\varepsilon)$. With Jensen’s inequality we eventually find $\langle \alpha^{(\ell+1)}, \mu \rangle \leq C$ for $\ell \geq 0$.

Combining this with (4.2) we obtain $n \leq 1 + \frac{C}{\varepsilon(1-q^{(n)})}$. So, as long as $q^{(n)} < q_{\text{target}}$ we have $n < 1 + \frac{C}{\varepsilon(1-q_{\text{target}})}$. By contraposition we know that there is some $n \leq 2 + \frac{C}{\varepsilon(1-q_{\text{target}})}$ such that $q^{(n)} \geq q_{\text{target}}$. \square

And finally, we formally establish convergence of the iterates.

COROLLARY 19 (Convergence of Asymmetric Algorithm). *Ignoring the stopping criterion, the iterates $(u^{(\ell)}, v^{(\ell)})$ of the asymmetric Algorithm 7 converge to a solution of the scaling problem and $q^{(\ell)} \rightarrow 1$.*

Proof. With the upper bound $\langle u^{(\ell)}, \mu \rangle \leq \exp(C/\varepsilon)$ (Proposition 18) we obtain the pointwise lower bound $v^{(\ell)}(y) \geq \exp(-C/\varepsilon)$ for all $\ell \geq 0$. Since $v^{(\ell)}$ is pointwise decreasing, it converges to some limit $v^{(\infty)} \geq \exp(-C/\varepsilon) > 0$.

The map $f : v^{(\ell)} \mapsto v^{(\ell+1)}$ is continuous for $v^{(\ell)} > 0$. With $v^{(\ell)} \rightarrow v^{(\infty)}$ and $v^{(\ell+1)} = f(v^{(\ell)}) \rightarrow v^{(\infty)}$ have $f(v^{(\infty)}) = v^{(\infty)}$ which implies that $v^{(\infty)}$ (together with the corresponding $u^{(\infty)} = \mu \oslash (K v^{(\infty)})$) solves the scaling problem. This implies convergence of $q^{(\ell)}$ to 1. \square

REMARK 5 (On the Stopping Criterion and Relation to [21]). *The criterion $q \geq q_{\text{target}}$ is motivated by Lemma 17, to provide a minimal increment of α during iterations. $1 - q$ measures the mass that is*

still missing and is equal to the L^1 error between the marginals of π and the desired marginals μ and ν . In pathological cases the dual variables (α, β) may still be far from optimizers, even though $q \geq q_{\text{target}}$ (see Example 1). In [21, Lemma 2] linear convergence of the marginals in the Hilbert projective metric is proven. This is a stricter measure of convergence, less prone to ‘premature’ termination. However, for small ε the contraction factor is roughly $1 - 4 \exp(-C/\varepsilon)$, which is impractical. The scaling $\mathcal{O}(1/\varepsilon)$ predicted by Proposition 18 is consistent with numerical observations when one uses the L^1 or L^∞ marginal error as stopping criterion (Sect. 5.2). Therefore we consider the q -criterion to be a reasonable measure for convergence, as long as one keeps $1 - q_{\text{target}} \ll \delta$ (Example 1).

EXAMPLE 1. We consider the 1×2 toy problem with the following parameters:

$$\mu = (1)^\top, \quad \nu = (1 - \delta \quad \delta)^\top, \quad c = (0 \quad C), \quad K = (1 - \delta \quad \delta \cdot e^{-C/\varepsilon})$$

for some $C > 0$, $\delta \in (0, 1)$ and some regularization strength $\varepsilon > 0$. And we consider the scaling factors (one for X , two choices for Y): $u = (1)^\top$, $v_1 = (1 \quad 1)^\top$, $v_2 = (1 \quad e^{C/\varepsilon})^\top$. Let $\pi_i = \text{diag}(u) K \text{diag}(v_i)$ and corresponding total masses q_i , $i = 1, 2$. We find:

$$\pi_1 = (1 - \delta \quad \delta \cdot e^{-C/\varepsilon}), \quad q_1 = 1 - \delta(1 - e^{-C/\varepsilon}), \quad \pi_2 = (1 - \delta \quad \delta), \quad q_2 = 1.$$

π_2 and $(\alpha, \beta_2) = \varepsilon \log(u, v_2)$ are primal and dual solutions. π_1 is sub-feasible (see Proposition 16). For fixed $\varepsilon > 0$, as $\delta \rightarrow 0$, q_1 tends to 1 (but is strictly smaller), i.e. the pair (u, v_1) has almost converged in the q -measure sense, but the distance between $\beta_1 = \varepsilon \log v_1$ and the actual solution β_2 is C .

REMARK 6 (Analogy to Auction Algorithm). For now assume $|X| = |Y| = N$ and μ, ν are normalized counting measures. Then line 4 in Algorithm 7, expressed in dual variables, becomes

$$\begin{aligned} \alpha(x) &\leftarrow \text{softmin}(\{c(x, y) - \beta(y) | y \in Y\}, \varepsilon) + \varepsilon \log N, \\ \hat{\beta}(y) &\leftarrow \text{softmin}(\{c(x, y) - \alpha(x) | x \in X\}, \varepsilon) + \varepsilon \log N. \end{aligned}$$

These are formally similar to the corresponding lines 7 and 13 in Algorithm 6. We can interpret the u -update in Algorithm 7 as x not just submitting a bid to the best candidate y , but to all candidates, weighted by the attractiveness (recall that in the Sinkhorn algorithm, a change in the dual variable directly implies a change in the primal iterate via (2.11)). Conversely, in line 5, y does not only accept the best bid, but bids from all candidates, again weighted by price. If there are too many bids (i.e. if $\hat{v}(y) < v(y)$), $\beta(y)$ decreases and thereby rejects superfluous offers.

Consequently, in Algorithm 7 one can observe that points in X compete for the mass in Y in a way similar to the auction algorithm by repeatedly increasing their prices until a different target seems more attractive or other competitors lose interest. In both algorithms the minimal increment is related to the parameter ε which leads to iteration bounds that are proportional to $1/\varepsilon$ (Props. 14 and 18). An attempt to mimic the analysis of ε -scaling is made in Section 4.4 (cf. Remark 9).

One can interpret the standard Sinkhorn algorithm with $v \leftarrow \hat{v}$ as y submitting a ‘counter-bid’ if it has not received enough bids. Such a symmetrization has also been discussed for the auction algorithm. But then the complexity analysis based on monotonous dual variables breaks down, and the algorithm may even run indefinitely (see ‘down iterations’ in [10]).

4.3. Stability of Dual Solutions. The main result of this section is Theorem 20, which provides stability of dual solutions to entropy regularized optimal transport (Def. 8) under changes of the regularization parameter ε . Its implications for ε -scaling are discussed in Sect. 4.4.

We consider a similar setup as in Sect. 4.2: $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$, $c \in \mathbb{R}_+^{X \times Y}$. Again, the reference measure ρ for regularization, see (2.8), is chosen to be the product measure $\rho(x, y) = \mu(x) \cdot \nu(y)$. For Theorem 20 we introduce an additional assumption on μ and ν . The necessity of this assumption can be demonstrated by counter-examples similar to Example 1.

ASSUMPTION 1 (Atomic Mass). For $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ there is some $M \in \mathbb{N}$ such that $\mu = r/M$, $\nu = s/M$ for $r \in \mathbb{N}^X$, $s \in \mathbb{N}^Y$.

THEOREM 20 (Stability of Dual Solutions under ε -Scaling). Let $\max\{|X|, |Y|\} \leq N < \infty$ and let μ and ν satisfy Assumption 1 for some $M \in \mathbb{N}$. For two regularization parameters $\varepsilon_1 > \varepsilon_2 > 0$, let (α_1, β_1)

and (α_2, β_2) be maximizers of the corresponding dual regularized optimal transport problems (Def. 8) and let $\Delta\alpha = \alpha_2 - \alpha_1$ and $\Delta\beta = \beta_2 - \beta_1$. Then

$$(4.3a) \quad \max \Delta\alpha - \min \Delta\alpha \leq \varepsilon_1 \cdot N \cdot (4 \log N + 24 \log M),$$

$$(4.3b) \quad \max \Delta\beta - \min \Delta\beta \leq \varepsilon_1 \cdot N \cdot (4 \log N + 24 \log M).$$

REMARK 7 (Relation to [16] and Motivation). [16] studies the convergence of entropy regularized linear programs to the unregularized variant and can be used to understand the limit of entropy regularized optimal transport (Def. 8). To apply [16], the constraint matrix must have full rank and the set of optimal solutions to (2.5b) must be bounded. When the cost c is finite this is achieved by arbitrarily fixing one dual variable, e.g. $\alpha(x_0) = 0$, and removing the corresponding column from the dual constraint matrix. The slight difference in the definition of the entropy (or the dual exponential barrier) can be absorbed into a change of variables which converges to the identity in the limit $\varepsilon \rightarrow 0$.

Then [16, Props. 3.1 and 3.2] imply that the optimal solutions of (2.19b) remain bounded and converge to a particular solution of (2.5b) as $\varepsilon \rightarrow 0$. Furthermore, [16] provides statements about the convergence of the optimal couplings (Prop. 4.1) and the asymptotic behaviour (Thm. 5.8).

The bounds derived in [16] depend on the geometry of the primal and dual feasible polytopes of (2.5), i.e. on the transport cost function c . In contrast, the bound of Thm. 20 does not depend on c . The motivation for deriving such a bound is the implication for ε -scaling, see Section 4.4. Note that Thm. 20 also implies that the optimal dual variables remain bounded as $\varepsilon \rightarrow 0$.

REMARK 8 (Proof Strategy). The proof requires several auxiliary definitions and lemmas. The estimate consists of two contributions: One stems from following paths within connected components of what we call assignment graph (defined in the following lemma), using the primal-dual relation (2.11). This reasoning is analogous to the proof strategy for ε -scaling in the auction algorithm (see [10]). However, between different connected components (2.11) is too weak to yield useful estimates. So a second contribution arises from a stability analysis of effective diagonal problems (in Lemmas 22 and 23).

LEMMA 21 (Assignment Graph). For two feasible couplings $\pi_1, \pi_2 \in \Pi(\mu, \nu)$ and a threshold $M^{-1} \geq 1$ the corresponding assignment graph is a bipartite directed graph with vertex sets (X, Y) and the set of directed edges

$$\mathcal{E} = \{(x, y) \in X \times Y : \pi_2(x, y) \geq \mu(x) \cdot \nu(y)/M\} \sqcup \{(y, x) \in Y \times X : \pi_1(x, y) \geq \mu(x) \cdot \nu(y)/M\}$$

where $(a, b) \in \mathcal{E}$ indicates a directed edge from $a \rightarrow b$.

The assignment graph has the following properties:

- (i) Every node has at least one incoming and one outgoing edge.
- (ii) Let $X_0 \subset X, Y_0 \subset Y$ such that there are no outgoing edges from (X_0, Y_0) to the rest of the vertices, then $|\mu(X_0) - \nu(Y_0)| < 1/M$. This is also true when there are no incoming edges from the rest of the vertices. If μ and ν are atomic, with atom size $1/M$ (see Assumption 1), then $\mu(X_0) = \nu(Y_0)$.
- (iii) Let μ and ν be atomic, with atom size $1/M$. Let $\{(X_i, Y_i)\}_{i=1}^R$ be the vertex sets of the strongly connected components of the assignment graph, for some $R \in \mathbb{N}$ (taking into account the orientation of the edges). Then the sets $\{X_i\}_{i=1}^R$ and $\{Y_i\}_{i=1}^R$ are partitions of X and Y , and $\mu(X_i) = \nu(Y_i)$ for $i = 1, \dots, R$.

Proof. Assume, a node $x \in X$ had no outgoing edge. Then $\sum_{y \in Y} \pi_2(x, y) < \mu(x)/M \leq \mu(x)$. This contradicts $\pi_2 \in \Pi(\mu, \nu)$. Existence of incoming edges follows analogously.

Let $\hat{X}_0 = X \setminus X_0, \hat{Y}_0 = Y \setminus Y_0$. If (X_0, Y_0) has no outgoing edges, then

$$\sum_{\substack{x \in X_0, \\ y \in \hat{Y}_0}} \pi_2(x, y) < \sum_{\substack{x \in X_0, \\ y \in \hat{Y}_0}} \mu(x) \cdot \nu(y)/M \leq \frac{1}{M}, \quad \sum_{\substack{x \in \hat{X}_0, \\ y \in Y_0}} \pi_1(x, y) < \sum_{\substack{x \in \hat{X}_0, \\ y \in Y_0}} \mu(x) \cdot \nu(y)/M \leq \frac{1}{M}.$$

Since $\pi_1, \pi_2 \in \Pi(\mu, \nu)$, the first inequality implies $\mu(X_0) = \pi_2(X_0 \times Y) = \pi_2(X_0 \times Y_0) + \pi_2(X_0 \times \hat{Y}_0) < \nu(Y_0) + 1/M$ and the second inequality implies $\nu(Y_0) < \mu(X_0) + 1/M$, i.e. $|\mu(X_0) - \nu(Y_0)| < 1/M$. With Assumption 1 for atom size $1/M$, this implies $\mu(X_0) = \nu(Y_0)$. The statement about incoming edges follows from $\mu(\hat{X}_0) = 1 - \mu(X_0)$ and $\nu(\hat{Y}_0) = 1 - \nu(Y_0)$.

Every node in (X, Y) is part of at least one strongly connected component (containing at least the node itself). If two strongly connected components have a common element, they are identical. Hence, the strongly connected components form partitions of X and Y . For some $x \in X$ (or $y \in Y$), let $X_{\text{out}} \subset X$ and $Y_{\text{out}} \subset Y$ be the set of nodes that can be reached from x , let $X_{\text{in}} \subset X$ and $Y_{\text{in}} \subset Y$ be the set of nodes from which one can reach x and let $(X_{\text{con}} = X_{\text{out}} \cap X_{\text{in}}, Y_{\text{con}} = Y_{\text{out}} \cap Y_{\text{in}})$ be the strongly connected component of x . Clearly $(X_{\text{out}}, Y_{\text{out}})$ has no outgoing edges. Hence, by (ii) one has $\mu(X_{\text{out}}) = \nu(Y_{\text{out}})$. Moreover, $(X_{\text{out}} \setminus X_{\text{in}}, Y_{\text{out}} \setminus Y_{\text{in}})$ has no outgoing edges, hence $\mu(X_{\text{out}} \setminus X_{\text{in}}) = \nu(Y_{\text{out}} \setminus Y_{\text{in}})$, from which follows that $\mu(X_{\text{con}}) = \nu(Y_{\text{con}})$. \square

LEMMA 22 (Reduction to Effective Diagonal Problem). *Let $\{X_i\}_{i=1}^R$ and $\{Y_i\}_{i=1}^R$ be partitions of X and Y , for some $R \in \mathbb{N}$, with $\mu(X_i) = \nu(Y_i)$ for $i = 1, \dots, R$. Let $\{y_i\}_{i=1}^R \subset Y$ such that $y_i \in Y_i$. Let $(\alpha^\dagger, \beta^\dagger)$ be optimizers for the dual entropy regularized optimal transport problem (Def. 8) for a regularization parameter $\varepsilon > 0$.*

Consider the following functional over \mathbb{R}^R :

$$\hat{J} : \mathbb{R}^R \rightarrow \mathbb{R}, \quad \hat{\beta} \mapsto -\varepsilon \sum_{i,j=1}^R \exp\left(-\frac{1}{\varepsilon} [d(i,j) + \hat{\beta}(i) - \hat{\beta}(j)]\right)$$

where $d \in \mathbb{R}^{R \times R}$ with

$$(4.4) \quad d(i,j) = -\varepsilon \log \left(\sum_{\substack{x \in X_i \\ y \in Y_j}} \exp\left(-\frac{1}{\varepsilon} [c(x,y) - \alpha^\dagger(x) - \beta^\dagger(y_i) - \beta^\dagger(y) + \beta^\dagger(y_j)]\right) \cdot \mu(x) \cdot \nu(y) \right).$$

Then $\hat{\beta}^\dagger \in \mathbb{R}^R$, given by $\hat{\beta}^\dagger(i) = \beta^\dagger(y_i)$, is a maximizer of \hat{J} . Conversely, if $\hat{\beta}^{\dagger\dagger}$ is a maximizer of \hat{J} , then there is a constant $b \in \mathbb{R}$, such that $\hat{\beta}^{\dagger\dagger}(i) = \hat{\beta}^\dagger(i) + b$ for all $i \in 1, \dots, R$.

Proof. We define the functional $\hat{J} : \mathbb{R}^R \rightarrow \mathbb{R}$ as follows:

$$\hat{J} : \hat{\beta} \mapsto J \left(\begin{pmatrix} \tilde{\alpha} \\ \tilde{\beta} \end{pmatrix} + \begin{pmatrix} -B_X \\ B_Y \end{pmatrix} \hat{\beta} \right)$$

where J denotes the dual functional of entropy regularized optimal transport (2.19b), and

- $\tilde{\alpha} \in \mathbb{R}^X$ with $\tilde{\alpha}(x) = \alpha^\dagger(x) + \beta^\dagger(y_i)$ when $x \in X_i$;
- $\tilde{\beta} \in \mathbb{R}^Y$ with $\tilde{\beta}(y) = \beta^\dagger(y) - \beta^\dagger(y_j)$ when $y \in Y_j$;
- $B_X \in \mathbb{R}^{X \times R}$ with $B_X(x, i) = 1$ if $x \in X_i$ and 0 else;
- $B_Y \in \mathbb{R}^{Y \times R}$ with $B_Y(y, j) = 1$ if $y \in Y_j$ and 0 else.

Then one has

$$\begin{pmatrix} \alpha^\dagger \\ \beta^\dagger \end{pmatrix} = \begin{pmatrix} \tilde{\alpha} \\ \tilde{\beta} \end{pmatrix} + \begin{pmatrix} -B_X \\ B_Y \end{pmatrix} \hat{\beta}^\dagger.$$

Since maximizing \hat{J} corresponds to maximizing J over an affine subspace, clearly $\hat{\beta}^\dagger$ is a maximizer of \hat{J} . Since \hat{J} inherits the invariance of J under constant shifts, any $\hat{\beta}^{\dagger\dagger}$ of the form given above, is also a maximizer. Consequently, we may add the constraint $\hat{\beta}^\dagger(1) = 0$, which does not change the optimal value. With this added constraint the functional becomes strictly convex, which implies a unique optimizer. Hence, any optimizer of the unconstrained functional can be written in the form of $\hat{\beta}^\dagger$.

Let us now give a more explicit expression of $\hat{J}(\hat{\beta})$. We find

$$\begin{aligned} \hat{J}(\hat{\beta}) &= \left\langle B_Y^\top \nu - B_X^\top \mu, \hat{\beta} \right\rangle - \varepsilon \sum_{i,j=1}^R \sum_{\substack{x \in X_i \\ y \in Y_j}} \exp\left(-\frac{1}{\varepsilon} [c(x,y) - \tilde{\alpha}(x) + \hat{\beta}(i) - \tilde{\beta}(y) - \hat{\beta}(j)]\right) \cdot \mu(x) \cdot \nu(y) \\ &\quad + \langle \mu, \tilde{\alpha} \rangle + \langle \nu, \tilde{\beta} \rangle + \varepsilon \cdot K(X \times Y). \end{aligned}$$

Note that the second line is constant w.r.t. $\hat{\beta}$. Since $\mu(X_i) = \nu(Y_i)$ the linear term vanishes and we can write $\hat{J}(\hat{\beta}) = -\varepsilon \sum_{i,j=1}^R \exp\left(-\frac{1}{\varepsilon} [d(i,j) + \hat{\beta}(i) - \hat{\beta}(j)]\right) + \text{const}$ with coefficients $d \in \mathbb{R}^{R \times R}$, as given above. The constant offset does not affect minimization. \square

LEMMA 23 (Effective Diagonal Problem and Stability). *For a parameter $\varepsilon > 0$ and a real matrix $d \in \mathbb{R}^{R \times R}$ consider the following functional:*

$$(4.5) \quad \hat{J}_{\varepsilon,d}(\beta) = \sum_{i,j=1}^R \exp([-d(i,j) - \beta(i) + \beta(j)]/\varepsilon)$$

Minimizers of $\hat{J}_{\varepsilon,d}$ exist.

Let $\varepsilon_1 \geq \varepsilon_2 > 0$ be two parameters and $d_1, d_2 \in \mathbb{R}^{R \times R}$ two real matrices. Let β_1^\dagger and β_2^\dagger be minimizers of $\hat{J}_{\varepsilon_1,d_1}$ and $\hat{J}_{\varepsilon_2,d_2}$, let $\Delta d = d_2 - d_1$, $\Delta\beta = \beta_2^\dagger - \beta_1^\dagger$. Let the matrix $w \in \mathbb{R}^{R \times R}$ be given by $w(i,j) = \max\{-\Delta d(i,j), \Delta d(j,i)\}$. Then $\max \Delta\beta - \min \Delta\beta \leq \text{maxdiam}(w) + 2\varepsilon_1 R \log R$, where

$$\text{maxdiam}(w) = \max \left\{ \sum_{i=1}^{k-1} w(j_i, j_{i+1}) : k \in \{2, \dots, R\}, j_i \in \{1, \dots, R\} \text{ for } i = 1, \dots, k, \text{ all } j_i \text{ distinct.} \right\}.$$

That is, $\text{maxdiam}(w)$ is the length of the longest cycle-less path in $\{1, \dots, R\}$ with edge lengths w .

The proofs of Theorem 20 and Lemma 23 can be found in Appendix A.

4.4. Application To ε -Scaling. Assuming that we know the dual solution for some $\varepsilon_1 > 0$, then Theorem 20 allows to bound the number of iterations of Algorithm 7 for some smaller $\varepsilon_2 \in (0, \varepsilon_1)$, independently of bounds on the cost function c . This may have implications for the efficiency of ε -scaling (see Remark 9).

PROPOSITION 24 (Single ε -Scaling Step). *Consider the set-up of Theorem 20. In particular, let $\varepsilon_1 > \varepsilon_2 > 0$ be two regularization parameters, let $(\alpha_1, \beta_1), (\alpha_2, \beta_2)$ be corresponding optimizers of (2.19b). If Algorithm 7 is initialized with $v^{(0)} = \exp(\beta_1/\varepsilon_2)$, with regularization ε_2 , and for a given $q_{\text{target}} \in (0, 1)$, the number of iterations n necessary to achieve $q^{(n)} \geq q_{\text{target}}$ is bounded by*

$$(4.6) \quad n \leq 2 + \frac{\varepsilon_1 N \cdot (4 \log N + 24 \log M) + \log M}{\varepsilon_2 (1 - q_{\text{target}})}.$$

Proof. For the optimal scaling factor u_1 of the ε_1 -problem we find:

$$u_1(x) := \exp(\alpha_1(x)/\varepsilon_1) = \left(\sum_{y \in Y} \exp\left(-\frac{1}{\varepsilon_1} [c(x,y) - \beta_1(y)]\right) \nu(y) \right)^{-1}$$

This implies $u_1(x)^{-1} \nu(y)^{-1} \geq \exp\left(-\frac{1}{\varepsilon_1} [c(x,y) - \beta_1(y)]\right)$ for all $(x,y) \in X \times Y$. With this we can bound the first iterate of the ε_2 -run of the algorithm by:

$$u^{(1)}(x) = \left(\sum_{y \in Y} \exp\left(-\frac{1}{\varepsilon_2} [c(x,y) - \beta_1(y)]\right) \nu(y) \right)^{-1} \geq \left(\sum_{y \in Y} (u_1(x) \nu(y))^{-\frac{\varepsilon_1}{\varepsilon_2}} \nu(y) \right)^{-1} \geq \left(\frac{u_1(x)}{M} \right)^{\frac{\varepsilon_1}{\varepsilon_2}}$$

where we have used $\nu(y) \geq 1/M$, Assumption 1. Eventually we find $\alpha^{(1)}(x) \geq \alpha_1(x) - \varepsilon_1 \log M$.

By monotonicity of the iterates we have $\beta_2 \leq \beta^{(\ell)} \leq \beta^{(0)} = \beta_1$ and $\beta^{(\ell)}(y') = \beta_1(y')$ for a suitable $y' \in Y$ (Proposition 14). Consequently $\max \Delta\beta = 0$. Then, from Theorem 20, we obtain $\beta_2(y) - \beta_1(y) \geq \min \Delta\beta \geq -\varepsilon_1 \cdot A$ where $A = N \cdot (4 \log N + 24 \log M)$. With this we can bound the u -iterates:

$$\begin{aligned} u^{(\ell)}(x) &\leq u_2(x) := \exp(\alpha_2(x)/\varepsilon_2) = \left(\sum_{y \in Y} \exp\left(-\frac{1}{\varepsilon_2} [c(x,y) - \beta_2(y)]\right) \nu(y) \right)^{-1} \\ &\leq \left(\sum_{y \in Y} \exp\left(-\frac{1}{\varepsilon_2} [c(x,y) - \beta_1(y)]\right) \nu(y) \right)^{-1} \exp\left(\frac{\varepsilon_1}{\varepsilon_2} A\right) \end{aligned}$$

With convexity of $s \mapsto s^{\varepsilon_1/\varepsilon_2}$ and Jensen's inequality we get

$$u^{(\ell)}(x) \leq \left(\sum_{y \in Y} \exp\left(-\frac{1}{\varepsilon_1}[c(x, y) - \beta_1(y)]\right) \nu(y) \right)^{-\varepsilon_1/\varepsilon_2} \exp\left(\frac{\varepsilon_1}{\varepsilon_2} A\right) = u_1(x)^{\varepsilon_1/\varepsilon_2} \exp\left(\frac{\varepsilon_1}{\varepsilon_2} A\right)$$

and finally $\alpha^{(\ell)}(x) \leq \alpha_1(x) + \varepsilon_1 A$. We summarize: $\alpha^{(\ell)}(x) - \alpha^{(0)}(x) \leq \varepsilon_1 (A + \log M)$. Now using Lemma 17 and arguing as in Proposition 18, we find that there is some $n \leq 2 + \frac{\varepsilon_1 (A + \log M)}{\varepsilon_2 (1 - q_{\text{target}})}$ such that $q^{(n)} \geq q_{\text{target}}$. \square

Let now $C = \max c$ for a cost function $c \geq 0$, let $\hat{\varepsilon} > 0$ be the desired final regularization parameter, pick some $\lambda \in (0, 1)$ and let $k \in \mathbb{N}$ such that $\hat{\varepsilon} \cdot \lambda^{-k} \geq C$. Let $\mathcal{E} = (\hat{\varepsilon} \cdot \lambda^{-k}, \hat{\varepsilon} \cdot \lambda^{-k+1}, \dots, \hat{\varepsilon})$ be a list of decreasing regularization parameters.

REMARK 9. Now we combine Algorithm 7 with ε -scaling, (cf. Algorithm 3). For $\varepsilon = \hat{\varepsilon} \cdot \lambda^{-k} \geq C$, according to Proposition 18 it will take at most $2 + \frac{1}{1 - q_{\text{target}}}$ iterations. It is tempting to deduce from Proposition 24 that for each subsequent value of ε at most $2 + \frac{A}{\lambda(1 - q_{\text{target}})}$ iterations are required, with $A = N \cdot (4 \log N + 24 \log M) + \log M$. For $N > 1$ the total number of iterations would then be bounded by $(2 + \frac{A}{\lambda(1 - q_{\text{target}})}) \cdot (k + 1)$. For fixed λ the step parameter k scales like $\log(C/\hat{\varepsilon})$. Consequently, the total number of iterations would be bounded by $\mathcal{O}(\log(C/\hat{\varepsilon}))$ w.r.t. the cost function and regularization, which would be analogous to ε -scaling for the auction algorithm (Remark 4).

There is an obvious gap in this reasoning: Theorem 20 assumes that β_1 is known exactly, while Algorithm 7 only provides an approximate result. From Example 1 we learn that in extreme cases this difference can be substantial and disrupt the efficiency of ε -scaling. Thus, additional assumptions on the problem are required to make the above argument rigorous.

However, as discussed in Remark 5, in practice we usually observe that approximate iterates are sufficient and we can therefore hope that ε -scaling does indeed serve its purpose.

5. Numerical Examples. Now we present a series of numerical experiments to confirm the usefulness of the modifications proposed in Sect. 3. We show that runtime and memory usage are reduced substantially. At the same time the adapted algorithm is still as versatile as the basic version of [14], Algorithm 1. But Algorithm 5 can solve larger problems at lower regularization, yielding very sharp results. We give examples for unbalanced transport, barycenters and Wasserstein gradient flows. The code used for the numerical experiments is available from the author's website.¹

5.1. Setup. We transport measures on $[0, 1]^d$ for $d \in \{1, 2, 3\}$, represented by discrete equidistant Cartesian grids. The distance between neighbouring grid points is denoted by h . For the squared Euclidean distance cost function $c(x, y) = |x - y|^2$, $x, y \in \mathbb{R}^d$, K is a Gaussian kernel with approximate width $\sqrt{\varepsilon}$. Therefore, it is useful to measure ε in units of h^2 . For $\varepsilon = h^2$ the blur induced by the entropy smoothing is on the length scale of one pixel. With the enhanced scaling algorithm we solve most problems in this section with $\varepsilon = 0.1 \cdot h^2$, leaving very little blur and giving a good approximation of the original unregularized problem (see Fig. 4).

Unless stated otherwise, we use the following settings: Test measures are mixtures of Gaussians, with randomized means and variances. The cost function is the squared Euclidean distance. ρ is the product measure $\mu \otimes \nu$ for optimal transport problems and the discretized Lebesgue measure on the product space for problems with variable marginals. For standard optimal transport the stopping criterion is the L^∞ error between prescribed marginals (μ, ν) and marginals of the primal iterate π (and likewise for Wasserstein barycenters). For all other models the primal-dual gap is used. We set $\theta = 10^{-20}$ for truncating the kernel and $\tau = 10^2$ as upper bound for (\tilde{u}, \tilde{v}) (cf. (3.9), Algorithm 2, line 4), implying a bound of $10^{-16} \cdot \rho(X \times Y)$ for the truncation error, which is many orders of magnitude below prescribed marginal accuracies or primal-dual gaps. The hierarchical partitions in the coarse-to-fine scheme are 2^d -trees, where each layer i is a coarser d -dimensional grid with grid constant h_i . For combination with ε -scaling (Algorithm 5) we choose the lists \mathcal{E}_i , $i > 0$, such that for the smallest ε_i in each \mathcal{E}_i we have roughly $\varepsilon_i/h_i^2 \approx 1$. On the finest scale, we go down to the desired final value of ε . All reported run-times were obtained on a single core of an Intel Xeon E5-2697 processor at 2.7 GHz.

¹<https://github.com/bernhard-schmitzer>

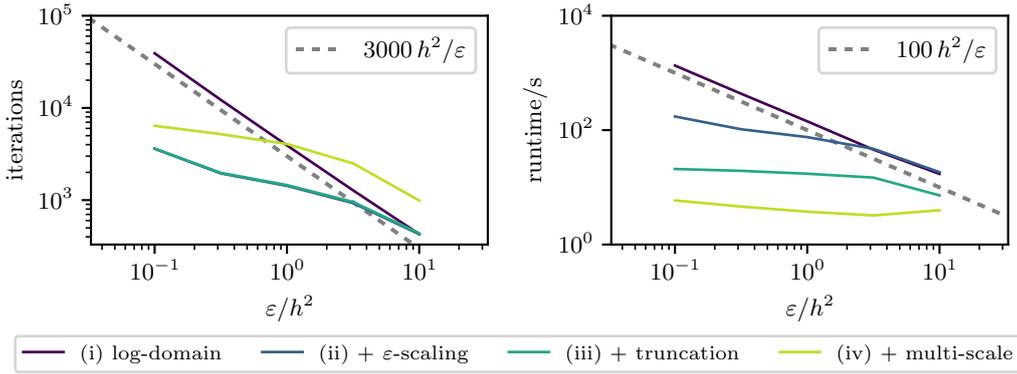


FIG. 2. *Efficiency of enhancements: average number of iterations and runtime for different ε and algorithms. $X = Y$ are 2-d 64×64 grids. (i) log-domain stabilized, Algorithm 2, (ii) with ε -scaling, Algorithm 3, (iii) with sparse stabilized kernel, (3.9), (iv) with multi-scale scheme, Algorithm 5. (ii) and (iii) need same number of iterations, but the sparse kernel requires less time. The naive implementation, Algorithm 1, requires same number of iterations as (i), but numerical overflow occurs at approximately $\varepsilon \leq 3h^2$.*

5.2. Efficiency of Enhanced Algorithm. The numerical efficiency of the subsequent modifications presented in Sect. 3, applied to the standard Sinkhorn algorithm, is illustrated in Fig. 2. While the stabilized algorithm (i) is not yet faster than the naive implementation, it can robustly solve the problem for all given values of ε . The required number of iterations scales like $\mathcal{O}(1/\varepsilon)$, in good agreement with the complexity analysis of Sect. 4.2. With ε -scaling (ii) the number of iterations is decreased substantially. Replacing the dense kernel with the adaptive truncated sparse kernel (iii) does not change the number of required iterations, but saves time and memory. With the multi-scale scheme the required number of iterations is slightly increased, since the initial dual variables obtained at a coarser level are only approximate solutions. But by reducing the number of variables during the early ε -scaling stages, the runtime is further decreased (cf. Fig. 1). The combination of all modifications leads to an average total speed-up of more than two orders of magnitude on this problem type.

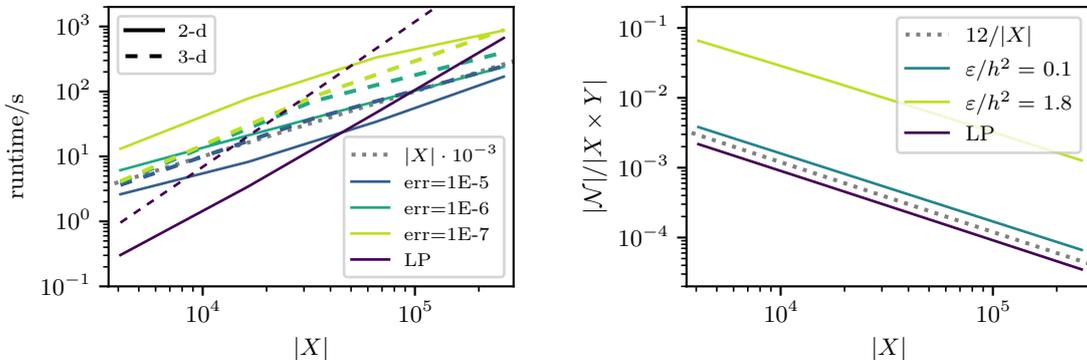


FIG. 3. *Average runtime and sparsity of Algorithm 5 for transporting test-images of different size (up to 512^2 pixels for 2-d, 64^3 for 3-d). Stopping criterion: L^∞ -marginal error, for different accuracy limits, final $\varepsilon = 0.1 \cdot h^2$. Performance of the adaptive sparse linear programming solver [40] given for comparison (LP). As expected, runtime increases with required accuracy. The runtime of the scaling algorithm scales more favourably (approximately linear) with $|X|$ and is competitive for large instances. The number of variables scales as $\mathcal{O}(1/|X|)$, suggesting that the number of variables per $x \in X$ is roughly constant. For the final $\varepsilon = 0.1 \cdot h^2$, the sparsity of the truncated kernel is comparable to [40]. For $\varepsilon = 1.8 \cdot h^2$, the largest value in \mathcal{E}_0 (the list for the finest scale), more variables are required.*

A runtime benchmark and study of the sparsity of the truncated kernel are given in Fig. 3. The runtime scales approximately linear with $|X|$ and for large problems the algorithm becomes faster than the adaptive sparse linear programming solver [40]. The final number of variables in the sparse kernel is comparable with the number of variables in [40], for higher values of ε , during scaling, more memory is required (cf. Fig. 4). This underlines again the importance of the coarse-to-fine scheme (Sect. 3.4).

It should be noted, that Fig. 2 shows results for 64×64 images, the smallest image size in Fig. 3. For larger images the runtime difference between (i-iv) would be even larger, but due to time and memory constraints, only (iv) can be run practically.

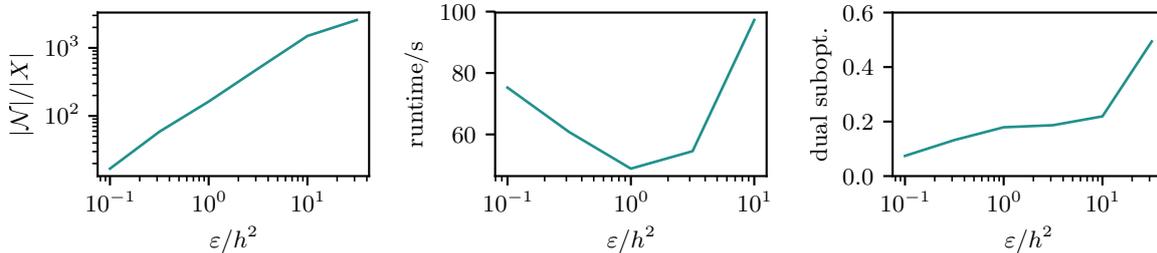


FIG. 4. Different final values for ε in Algorithm 5. $X = Y$ are 2-d 256×256 grids. **Left** Average number of variables in truncated kernel per $x \in X$. For $\varepsilon = 0.1 \cdot h^2$ only about 10 variables per $x \in X$ are required. As expected, this number increases with ε (cf. Fig. 1). **Center** For large ε , the runtime decreases with ε , since the number of variables decreases (cf. left plot). For smaller ε , the runtime increases again, since more stages of ε -scaling are required. **Right** The optimal regularized dual variables were transformed into feasible unregularized dual variables, by decreasing each $\alpha(x)$ until all dual constraints $\alpha(x) + \beta(y) \leq c(x, y)$ were met, (2.5b). The sub-optimality of these dual variables is shown. As expected (see Sect. 1.1) they converge towards a dual optimizer. Absolute optimal value was between 100 and 400 for the used test problems, i.e. for small ε , sub-optimality is small compared to total scale.

The impact of different final values for ε is outlined in Fig. 4. As expected, the number of variables in the truncated kernel increases with ε . This leads to two competing trends in the overall runtime: For large ε , the kernel truncation is less efficient, leading to an increase with ε . For small ε , the number of variables is very small, but more and more stages of ε -scaling are necessary, increasing the runtime as ε decreases further. Convergence of the regularized optimal dual variables to the unregularized optimal duals is exemplified in the right panel, justifying the use of the approximate entropy regularization technique for transport-type problems. While one may consider the dual sub-optimality at $\varepsilon \approx 30 h^2$ sufficiently accurate, we point out that the corresponding primal coupling still contains considerable blur (cf. Fig. 1) and that due to less sparsity the runtime is actually higher than for $\varepsilon \approx h^2$.

As illustrated by Figs. 3 and 4, by choosing the threshold for the stopping criterion and the desired final ε , one can tune between required precision and available runtime.

REMARK 10 (Interplay of Modifications). *The numerical findings presented in Figs. 2-4 underline how each of the modifications discussed in Sect. 3 builds on the previous ones and that all four of them are required for an efficient algorithm. The log-domain stabilization is an indispensable prerequisite for running the scaling algorithms with small regularization. However, for small ε , convergence tends to become extremely slow (cf. Fig. 2), therefore ε -scaling is needed to reduce the number of iterations. For small ε , kernel truncation significantly reduces the number of variables and accelerates the algorithm (cf. Figs. 2 and 4). However, for large ε (which must be passed during ε -scaling), far fewer variables are truncated and the algorithm cannot be run on large problems. This can be avoided by using the coarse-to-fine scheme, completing the algorithm. In principle it is possible, only to combine log-domain stabilization with kernel truncation, and to skip ε -scaling and the coarse-to-fine scheme. While this tends to solve the stability and memory issues, convergence is still impractically slow.*

5.3. Versatility. The framework of scaling algorithms developed in [14], see Sect. 2, allows to solve more general transport-type problems for which the enhancements of Sect. 3 still apply. We now give some examples to demonstrate this flexibility. The scope of the following examples is similar to [14], but with Algorithm 5 one can solve larger problems with smaller regularization.

KL Fidelity and Wasserstein-Fisher-Rao distance. For the marginal function $F_X(\sigma) = \lambda \cdot \text{KL}_X(\sigma|\mu)$ with a given reference measure $\mu \in \mathbb{R}_+^X$ and a weight $\lambda > 0$, see Def. 6, one obtains for the (stabilized) proxdiv operator

$$(5.1) \quad \text{proxdiv}_\varepsilon F_X(\sigma) = (\mu \odot \nu)^{\frac{\lambda}{\lambda+\varepsilon}}, \quad \text{proxdiv}_\varepsilon F_X(\sigma, \alpha) = \exp\left(-\frac{\alpha}{\lambda+\varepsilon}\right) \odot (\mu \odot \nu)^{\frac{\lambda}{\lambda+\varepsilon}}.$$

A proof is given in [14]. Compared to the standard Sinkhorn algorithm, the only modification is the pointwise power of the iterates. As $\lambda \rightarrow \infty$ the Sinkhorn iterations are recovered. In the stabilized oper-

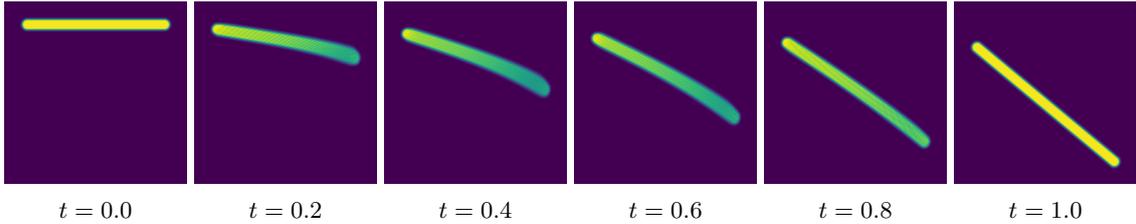


FIG. 5. Geodesic for Wasserstein-Fisher-Rao distance on $[0, 1]^2$, approximated by a 256×256 grid, computed as barycenters between end-points with varying weights. Mass that travels further, is decreased during transport to save cost.

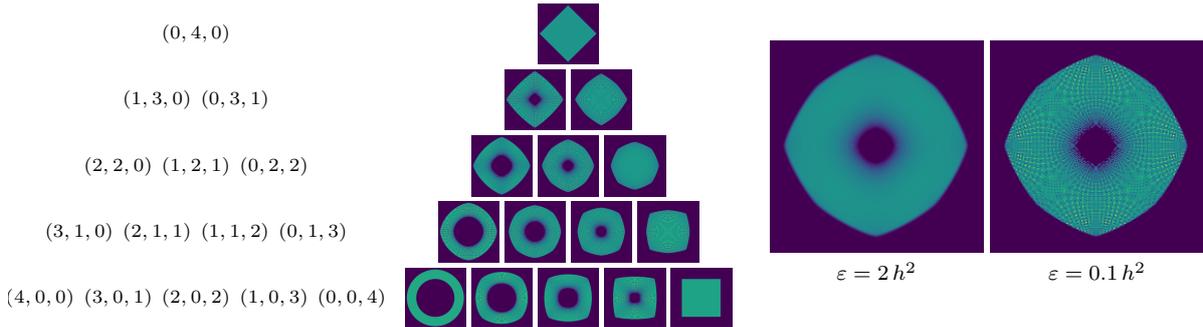


FIG. 6. Barycenters in Wasserstein space over $[0, 1]^2$, computed on 256×256 grids for $\varepsilon = 0.1 h^2$. **Left** Weights $4 \cdot (\lambda_1, \lambda_2, \lambda_3)$ for shown barycenters. **Center** ‘Barycentric triangle’ spanned by a ring, a diamond and a square for weights on the left. **Right** Close-up of the $\lambda = (1, 2, 1)/4$ barycenter for $\varepsilon = 2 h^2$ (as reported in [6]) and $\varepsilon = 0.1 h^2$, computed with the adapted algorithm. The $\varepsilon = 0.1 h^2$ version is much sharper, revealing discretization artifacts.

ator only the exponential $\exp\left(\frac{-\alpha}{\lambda+\varepsilon}\right)$ needs to be evaluated, which remains bounded as $\varepsilon \rightarrow 0$. Algorithm 5 performs similarly with KL-fidelity as with fixed marginal constraints, allowing to efficiently solve large unbalanced transport problems. Since the truncation scheme can also be used with non-standard cost functions such as (2.3), this includes in particular the Wasserstein-Fisher-Rao (WFR) distance. Fig. 5 shows a geodesic for the WFR distance, to intuitively illustrate its properties. The geodesic has been computed as weighted barycenters between its endpoints (see below). For a direct dynamic formulation we refer to [26, 13, 31]. For the relation to the KL soft-marginal formulation, Def. 6, see [31, 15].

Wasserstein barycenters. Wasserstein barycenters as a natural generalization of the Riemannian center of mass have been studied in [1]. The computation of entropy regularized Wasserstein barycenters with a Sinkhorn-type scaling algorithm has been presented in [6], an alternative numerical approach can be found in [19]. The iterations can be considered as a special case of the framework in [14]. Here, we very briefly recall the iterations. Derivations and proofs can be found in [6].

We want to compute the (entropy regularized) Wasserstein barycenter of a tuple $(\mu_1, \dots, \mu_n) \in \mathbb{R}^{X \times n}$ over a common base space $X = Y$ with metric d with non-negative weights $(\lambda_1, \dots, \lambda_n)$ that sum to one. The primal functional can be written as an optimization problem over a tuple $(\pi_i)_{i=1}^n = (\pi_1, \dots, \pi_n) \in \mathbb{R}^{(X \times X) \times n}$ of couplings, which requires a slight generalization of Def. 7, see [14]. It is given by

$$(5.2) \quad E((\pi_i)_i) = F_1((P_X \pi_i)_i) + F_2((P_Y \pi_i)_i) + \sum_{i=1}^n \lambda_i \text{KL}(\pi_i | K)$$

where

$$F_1((\nu_i)_i) = \sum_{i=1}^n \nu_{\{\mu_i\}}(\nu_i), \quad F_2((\nu_i)_i) = \begin{cases} 0 & \text{if } \exists \sigma \in \mathbb{R}^X \text{ s.t. } [\sigma = \nu_i \forall i = 1, \dots, n], \\ +\infty & \text{else.} \end{cases}$$

and K is the kernel (2.8) over $X \times X$ for the cost $c = d^2$. When an optimizer $(\pi_i^\dagger)_i$ is found, the common second marginal of all π_i^\dagger is the sought-after barycenter. To solve (5.2) one considers again a suitable dual

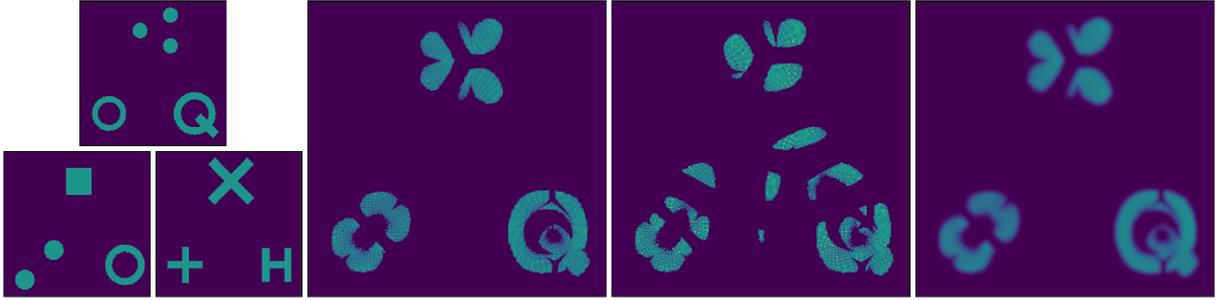


FIG. 7. Comparison of different barycenters models: **First column** Corner points of barycentric triangle. Each measure consists of three ‘groups’. **Second column** Wasserstein-Fisher-Rao barycenter for $\lambda = (1, 2, 1)/4$ for $\varepsilon = 0.1 h^2$. **Third column** Wasserstein barycenter between normalized reference measures for $\varepsilon = 0.1 h^2$. Unlike the ‘unbalanced’ barycenters, here mass must be transferred between the different ‘groups’ of the reference measures. **Fourth column** Gaussian Hellinger-Kantorovich barycenter for $\varepsilon = 6.55 h^2$, as computed with Gaussian convolution without log-domain stabilization.

problem and uses alternating optimization. Updates corresponding to F_1 decompose into independent standard Sinkhorn iterations for each marginal, the update for F_2 couples all marginals, see [6, 14]. The adaptations from Sect. 3 remain applicable. A barycentric triangle computed with Algorithm 5 is shown in Fig. 6. The log-domain stabilization allows to reach a lower final regularization ε as for example in [6]. Regularization can be made so small that discretization artifacts become visible. While this may not look entirely pleasing, it clearly gives a better approximation to the unregularized problem and illustrates that with log-domain stabilization entropy regularized numerical methods can produce sharp results.

Wasserstein-Fisher-Rao barycenters. Similarly one can define barycenters for transport distances with KL marginal fidelity, which includes the Gaussian Hellinger-Kantorovich (GHK) distance and the Wasserstein-Fisher-Rao (WFR) distance (Def. 6). The primal functional is given by (5.2) with

$$F_1((\nu_i)_i) = \Lambda \cdot \sum_{i=1}^n \lambda_i \text{KL}(\nu_i | \mu_i), \quad F_2((\nu_i)_i) = \inf_{\sigma \in \mathbb{R}^X} \Lambda \cdot \sum_{i=1}^n \lambda_i \text{KL}(\nu_i | \sigma),$$

where $\Lambda > 0$ is a global weight of the KL-fidelity. When a primal optimizer is found, the minimizing σ in F_2 yields the sought-after barycenter. We refer to [14] for details. Partial optimization corresponding to F_1 can again be done separately for each marginal, leading to KL fidelity updates as given by (5.1). The update corresponding to F_2 is again coupled [14], adaptations from Sect. 3 remain applicable.

Wasserstein Gradient Flows. In [36] diagonal scaling algorithms were extended to compute proximal steps for entropy regularized optimal transport to approximate gradient flows in Wasserstein space (cf. Sect. 1.1). This was then subsumed into the general framework of [14]. Here we give an example for the porous medium equation, for more details we refer to [36, 14]. Let

$$(5.3) \quad F : \mathcal{P}(X) \rightarrow \overline{\mathbb{R}}, \quad \mu \mapsto \sum_{x \in X} u \left(\frac{\mu(x)}{\mathcal{L}(x)} \right) \mathcal{L}(x) + \sum_{x \in X} v(x) \mu(x)$$

where $u(s) = s^2$, \mathcal{L} is the discretized Lebesgue measure on $X \subset \mathbb{R}^d$ and $v : X \rightarrow \overline{\mathbb{R}}$ is a potential. Then, for some initial $\mu^{(0)} \in \mathcal{P}(X)$ and a time step size $\tau > 0$ we iteratively construct a sequence $(\mu^{(\ell)})_\ell$ where $\mu^{(\ell+1)}$ is given by the proximal step of F with step size τ w.r.t. the entropy regularized Wasserstein distance on X from reference point $\mu^{(\ell)}$. Based on Def. 8, $\mu^{(\ell+1)}$ can be computed as follows:

$$(5.4) \quad \pi^{(\ell+1)} := \underset{\pi \in \mathcal{P}(X^2)}{\text{argmin}} \left(\iota_{\{\mu^{(\ell)}\}}(\text{P}_X \pi) + 2\tau \cdot F(\text{P}_Y \pi) + \varepsilon \text{KL}(\pi | K) \right), \quad \mu^{(\ell+1)} := \text{P}_Y \pi^{(\ell+1)},$$

where K is the kernel w.r.t. the squared Euclidean distance on X . Then introduce the time-continuous interpolation $\bar{\mu} : \mathbb{R}_+ \rightarrow \mathcal{P}(X)$, $t \mapsto \mu^{(\ell)}$ when $t \in [\tau \cdot \ell, \tau \cdot (\ell + 1))$. Consider now the limit $(\tau, \varepsilon) \rightarrow 0$ in a way such that $\varepsilon |\log \varepsilon| \leq \tau^2$. Then, up to discretization, the function $\bar{\mu}$ converges to a solution

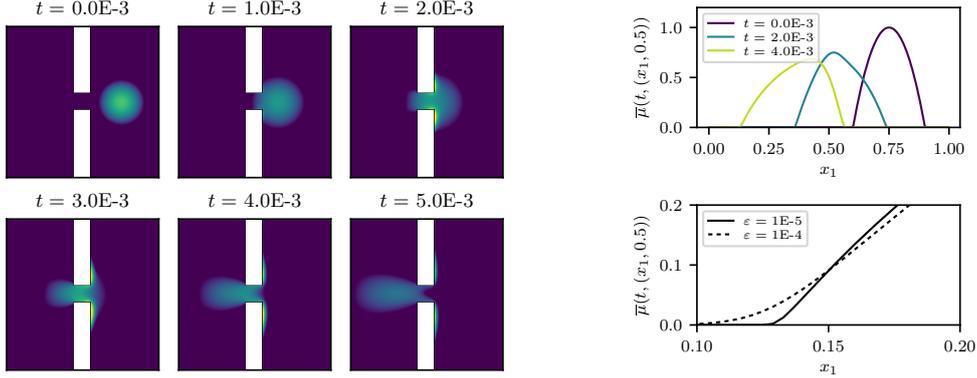


FIG. 8. **Left** Entropic Wasserstein gradient flow for the porous media equation on $[0, 1]^2$, approximated by a 256×256 grid with $\varepsilon = 10^{-5} \approx 0.66 h^2$, $\tau = 2 \cdot 10^{-4}$. The energy is given by (5.3) with $v((x_1, x_2)) = 100 \cdot x_1$ if $x = (x_1, x_2) \in \Omega$, $v(x) = +\infty$ otherwise and $\Omega = [0, 1]^2 \setminus \hat{\Omega}$ where $\hat{\Omega}$ is a ‘barrier’ indicated by the white rectangles. **Top Right** Cross section of density at different times along $x_2 = 0.5$. **Bottom Right** Close-up for $t = 4 \cdot 10^{-3}$ for different values of regularization ε . For $\varepsilon = 10^{-5}$ the compact support of $\bar{\mu}$, a characteristic feature of the porous media equation, is numerically well preserved. Without log-domain stabilization, for $\varepsilon = 10^{-4}$ the entropic blur quickly distorts this feature.

of the porous media PDE $\partial_t \bar{\mu} = \Delta(\bar{\mu}^2) + \text{div}(\bar{\mu} \cdot \nabla v)$. A proof is given in [12]. Problem (5.4) is an instance of Def. 7 and can be solved by alternating dual optimization [14]. A numerical example is shown in Fig. 8. As in the previous experiments, Algorithm 5 allows to use log-domain stabilization on large problems, producing sharp results. In this example, the compact support of the porous media equation is numerically well preserved.

6. Conclusion. Scaling algorithms for entropy regularized transport-type problems have become a wide-spread numerical tool. Naive implementations have some severe numerical limitations, in particular for small regularization and on large problems. In this article, we proposed an enhanced variant of the standard scaling algorithm to address these issues: Diverging scaling factors and slow convergence are remedied by log-domain stabilization and ε -scaling. Required runtime and memory are significantly reduced by adaptive kernel truncation and a coarse-to-fine scheme. A new convergence analysis for the Sinkhorn algorithm was developed. Numerical examples showed the efficiency of the enhanced algorithm, confirmed the scaling predicted by the convergence analysis and demonstrated that the algorithm can produce sharp results on a wide range of transport-type problems. Potential directions for future research are the more detailed study of ε -scaling, a more systematic understanding of the stability of the log-domain stabilization and application to multi-marginal problems.

Acknowledgements. L enaic Chizat, Luca Nenna and Gabriel Peyr e are thanked for stimulating discussions. Bernhard Schmitzer was supported by the European Research Council (project SIGMA-Vision).

Appendix A. Additional Proofs.

A.1. Proof of Lemma 23. First, we establish existence of minimizers. For some $\varepsilon > 0$, $d \in \mathbb{R}^{R \times R}$ the functional $\beta \mapsto \hat{J}_{\varepsilon, d}(\beta)$ is convex and bounded from below. Further, it is invariant under adding the same constant to all components of β . Hence, the optimal value $\min_{\beta} \hat{J}_{\varepsilon, d}(\beta)$ is not changed by adding the constraint $\beta(1) = 0$. With this added constraint the functional becomes strictly convex and coercive in the remaining variables, hence a unique minimizer exists. The full set of minimizers is then obtained via constant shifts.

The first order optimality condition for the functional yields for the i -th component of β :

$$\beta(i) = \frac{1}{2} \left[\text{softmax}_{j: j \neq i}(-d(i, j) + \beta(j), \varepsilon) + \text{softmin}_{j: j \neq i}(d(j, i) + \beta(j), \varepsilon) \right],$$

where the subscript $j : j \neq i$ denotes that softmax is taken only over components $\{1, \dots, R\} \setminus \{i\}$. Finiteness of d ensures that this expression is meaningful. Let $i_1 \in \{1, \dots, R\}$ be an index where $\Delta\beta$ is maximal, i.e. $\Delta\beta(i_1) = \max \Delta\beta$.

From the optimality conditions for $\beta_a(i_1)$, $a = 1, 2$, and (1.3) we obtain:

$$\begin{aligned}\beta_a^\dagger(i_1) &= \frac{1}{2} \left[\operatorname{softmax}_{j:j \neq i_1}(-d_a(i_1, j) + \beta_a^\dagger(j), \varepsilon_a) + \operatorname{softmin}_{j:j \neq i_1}(d_a(j, i_1) + \beta_a^\dagger(j), \varepsilon_a) \right], \\ \Delta\beta(i_1) &\leq \frac{1}{2} \left[\max_{j:j \neq i_1}(-\Delta d(i_1, j) + \Delta\beta(j)) + \max_{j:j \neq i_1}(\Delta d(j, i_1) + \Delta\beta(j)) + (\varepsilon_1 + \varepsilon_2) \cdot \log R \right] \\ &\leq \max_{j:j \neq i_1}(w(i_1, j) + \Delta\beta(j)) + \varepsilon_1 \log R\end{aligned}$$

where $w(i, j) = \max\{-\Delta d(i, j), \Delta d(j, i)\}$. This implies there is some $i_2 \in \{1, \dots, R\} \setminus \{i_1\}$ with

$$\Delta\beta(i_2) \geq \Delta\beta(i_1) - w(i_1, i_2) - \varepsilon_1 \cdot \log R.$$

We will call the index i_2 a child of i_1 . We now repeat this reasoning to derive lower bounds for other entries of $\Delta\beta$. For this we must ‘remove’ the index i_2 from the problem, defining a reduced problem. Let $I_1 = \{i_1, i_2\}$ and let $I_2 = \{1, \dots, R\} \setminus I_1$. We will keep all variables of β with indices in I_2 , but describe all variables with indices in I_1 by a single reduced variable. For this we consider vectors in $\mathbb{R}^{1+|I_2|}$, where we index the entries by $\{i_1\} \cup I_2$. One can think of this as a vector in \mathbb{R}^R , where we have ‘crossed out’ entries corresponding to I_1 and replaced them by a single effective entry, indexed with i_1 . For $a = 1, 2$ we consider the reduced functionals $\hat{\mathcal{J}}_a : \hat{\beta} \mapsto \hat{\mathcal{J}}_{\varepsilon_a, d_a}(\tilde{\beta}_a + B \hat{\beta})$ where $\tilde{\beta}_a \in \mathbb{R}^R$ is a constant offset, $\hat{\beta} \in \mathbb{R}^{1+|I_2|}$ is the reduced variable and $B \in \mathbb{R}^{R \times (1+|I_2|)}$ is a matrix that implements the parametrization. We set

$$\tilde{\beta}_a(j) = \begin{cases} \beta_a^\dagger(j) - \beta_a^\dagger(i_1) & \text{if } j \in I_1, \\ 0 & \text{else,} \end{cases} \quad B(j, k) = \begin{cases} 1 & \text{if } j \in I_1, k = i_1, \\ 1 & \text{if } j = k \in I_2, \\ 0 & \text{else.} \end{cases}$$

So the reduced functionals are given by

$$\begin{aligned}\hat{\mathcal{J}}_a(\hat{\beta}) &= \sum_{\substack{j \in I_1, \\ k \in I_1}} \exp([-d_a(j, k) - \tilde{\beta}_a(j) + \tilde{\beta}_a(k)]/\varepsilon_a) + \sum_{\substack{j \in I_1, \\ k \in I_2}} \exp([-d_a(j, k) - \tilde{\beta}_a(j) - \hat{\beta}(i_1) + \hat{\beta}(k)]/\varepsilon_a) \\ &\quad + \sum_{\substack{j \in I_2, \\ k \in I_1}} \exp([-d_a(j, k) - \hat{\beta}(j) + \tilde{\beta}_a(k) + \hat{\beta}(i_1)]/\varepsilon_a) + \sum_{\substack{j \in I_2, \\ k \in I_2}} \exp([-d_a(j, k) - \hat{\beta}(j) + \hat{\beta}(k)]/\varepsilon_a) \\ &= \sum_{\substack{j \in \{i_1\} \cup I_2, \\ k \in \{i_1\} \cup I_2}} \exp([-D_a(j, k) - \hat{\beta}(j) + \hat{\beta}(k)]/\varepsilon_a)\end{aligned}$$

with the reduced coefficient matrix $D_a \in \mathbb{R}^{(1+|I_2|)^2}$ with entries

$$D_a(j, k) = \begin{cases} \operatorname{softmin}_{r \in I_1, s \in I_1} (d_a(r, s) + \tilde{\beta}_a(r) - \tilde{\beta}_a(s), \varepsilon_a) & \text{if } j = i_1, k = i_1, \\ \operatorname{softmin}_{r \in I_1} (d_a(r, k) + \tilde{\beta}_a(r), \varepsilon_a) & \text{if } j = i_1, k \in I_2, \\ \operatorname{softmin}_{s \in I_1} (d_a(j, s) - \tilde{\beta}_a(s), \varepsilon_a) & \text{if } j \in I_2, k = i_1, \\ d_a(j, k) & \text{if } j \in I_2, k \in I_2. \end{cases}$$

Consider the reduced variables $\hat{\beta}_a^\dagger \in \mathbb{R}^{1+|I_2|}$ with entries

$$\hat{\beta}_a^\dagger(j) = \begin{cases} \beta_a^\dagger(i_1) & \text{if } j = i_1, \\ \beta_a^\dagger(j) & \text{if } j \in I_2. \end{cases}$$

Then $\beta_a^\dagger = \tilde{\beta}_a + B \hat{\beta}_a^\dagger$ and therefore $\hat{\beta}_a^\dagger$ are minimizers of $\hat{\mathcal{J}}_a$. Note also that $\hat{\beta}_2^\dagger(j) - \hat{\beta}_1^\dagger(j) = \Delta\beta(j)$ for $j \in \{i_1\} \cup I_2$. Using the optimality conditions for the reduced functionals and arguing as above, we find

$$\Delta\beta(i_1) \leq \max_{k \in I_2}(W(i_1, k) + \Delta\beta(k)) + \varepsilon_1 \log R$$

where $W(i_1, k) = \max\{-\Delta D(i_1, k), \Delta D(k, i_1)\}$ for $k \in I_2$ and $\Delta D = D_2 - D_1$. With (1.3) we find

$$\begin{aligned} -\Delta D(i_1, k) &\leq \max_{j \in I_1} (-\Delta d(j, k) - \Delta\beta(j) + \Delta\beta(i_1)) + \varepsilon_2 \log R, \\ \Delta D(k, i_1) &\leq \max_{j \in I_1} (\Delta d(k, j) - \Delta\beta(j) + \Delta\beta(i_1)) + \varepsilon_1 \log R \end{aligned}$$

and eventually $W(i_1, k) \leq \max_{j \in I_1} (w(j, k) - \Delta\beta(j)) + \Delta\beta(i_1) + \max\{\varepsilon_1, \varepsilon_2\} \cdot \log R$. So there is some index $i_3 \in I_2$ such that

$$\Delta\beta(i_3) \geq \min_{j \in I_1} (-w(j, i_3) + \Delta\beta(j)) - 2\varepsilon_1 \log R.$$

The index i_3 will be called a child of the minimizing index $j \in I_1$ on the r.h.s. (or one of the minimizing indices). Then we add i_3 to the set I_1 and repeat the argument with the reduced functional, to obtain an index i_4 and repeat this until I_1 contains all indices.

Since we assign every new index i_k that is added to I_1 as a child to one parent node in I_1 , this also constructs a tree graph with root node i_1 (finiteness of d and consequently D implies that this graph is connected). For an index i_k let (i_1, i_2, \dots, i_k) be the unique path from the root to i_k . Then

$$\Delta\beta(i_k) \geq -\sum_{j=2}^k w(i_{j-1}, i_j) + \Delta\beta(i_1) - 2(k-1)\varepsilon_1 \log R \geq -\max\text{diam}(w) + \Delta\beta(i_1) - 2\varepsilon_1 R \log R.$$

Since $\Delta\beta(i_1) = \max \Delta\beta$ the result follows.

A.2. Proof of Theorem 20. Let π_1, π_2 be the primal optimizers associated with (α_1, β_1) and (α_2, β_2) and consider the assignment graph for π_1 and π_2 and threshold $1/M$ (see Lemma 21). Let $\{(X_i, Y_i)\}_{i=1}^R$ be the strongly connected components of the assignment graph. By virtue of Lemma 21(iii), $\mu(X_i) = \nu(Y_i)$ for $i = 1, \dots, R$. Pick some representatives $\{y_i\}_{i=1}^R \subset Y$ such that $y_i \in Y_i$ for $i = 1, \dots, R$.

For $a = 1, 2$, let now \hat{J}_a be the reduced effective diagonal functionals, defined in Lemma 22, corresponding to spaces (X, Y) , marginals (μ, ν) , parameters ε_a , cost c , the partitions given by the strongly connected components and the representatives $\{y_i\}_{i=1}^R$. Let d_a be the corresponding effective coefficients (finite, since c is finite), let $\hat{\beta}_a^\dagger$ be two corresponding maximizers and let $\Delta d = d_2 - d_1$, $\Delta\hat{\beta} = \hat{\beta}_2^\dagger - \hat{\beta}_1^\dagger$. By virtue of Lemma 23 one has $\max \Delta\hat{\beta} - \min \Delta\hat{\beta} \leq \max\text{diam}(w) + 2\varepsilon^1 R \log R$, where $w \in \mathbb{R}^{R \times R}$ with $w(i, j) = \max\{-\Delta d(i, j), \Delta d(j, j)\}$.

Now we derive some estimates on Δd . Consider once more the assignment graph for π_1, π_2 and threshold $1/M$. For every edge $y \rightarrow x$ we have (using (2.11))

$$\alpha_1(x) + \beta_1(y) - c(x, y) \geq -\varepsilon_1 \log M.$$

Moreover, from the marginal conditions we find $\pi_2(x, y) \leq \nu(y)$, which implies

$$\alpha_2(x) + \beta_2(y) - c(x, y) \leq \varepsilon_2 \log M.$$

Combining the two estimates, we obtain $\Delta\alpha(x) + \Delta\beta(y) \leq (\varepsilon_1 + \varepsilon_2) \log M \leq 2\varepsilon_1 \log M := L$. Similarly, for edges $x \rightarrow y$ we obtain $\Delta\alpha(x) + \Delta\beta(y) \geq -(\varepsilon_1 + \varepsilon_2) \log M \geq -L$. Let now (y_1, x_1, \dots, y_k) be an alternating path in (X, Y) , then, by combining the above inequalities we find $\Delta\beta(y_{j+1}) \geq \Delta\beta(y_j) - 2 \cdot L$ for $j = 1, \dots, k-1$ and eventually

$$\Delta\beta(y_k) - \Delta\beta(y_1) \geq -2 \cdot (k-1) \cdot L.$$

Similarly, for a path $(x_1, y_2, x_2, \dots, y_k)$ get $\Delta\alpha(x_1) + \Delta\beta(y_k) \geq -(2k-1) \cdot L$, and for a path $(y_1, x_1, \dots, y_k, x_k)$ get $\Delta\alpha(x_k) + \Delta\beta(y_1) \leq (2k-1) \cdot L$.

Consider now a partition cell (X_i, Y_i) and let $y_i \in Y_i$ be the selected ‘representative’, as described above. For every $y \in Y_i$ there is a path to and from y_i with at most $2(|Y_i| - 1)$ edges, for every $x \in X_i$

there is a path to and from y_i with at most $2|Y_i| - 1$ edges. With $\Delta\tilde{\alpha}(x) = \Delta\alpha(x) + \Delta\beta(y_i)$ and $\Delta\tilde{\beta}(y) = \Delta\beta(y) - \Delta\beta(y_i)$ we therefore obtain

$$|\Delta\tilde{\alpha}(x)| \leq (2|Y_i| - 1) \cdot L, \quad |\Delta\tilde{\beta}(y)| \leq 2(|Y_i| - 1) \cdot L.$$

We recall (4.4)

$$d_a(i, j) = \operatorname{softmin}_{\substack{x \in X_i, \\ y \in Y_j}} \left(c(x, y) - \tilde{\alpha}_a(x) - \tilde{\beta}_a(y) - \varepsilon_a \log(\mu(x)\nu(y)), \varepsilon_a \right)$$

and get

$$\begin{aligned} \Delta d(i, j) &\leq \max_{\substack{x \in X_i, \\ y \in Y_j}} \left(-\Delta\tilde{\alpha}(x) - \Delta\tilde{\beta}(y) - \Delta\varepsilon \cdot \log(\mu(x)\nu(y)) \right) + \varepsilon_1 \cdot \log(|X_i||Y_j|) \\ &\leq 4|Y_i|L + \varepsilon_1 \cdot \log(|X_i||Y_j|) \\ \Delta d(i, j) &\geq \min_{\substack{x \in X_i, \\ y \in Y_j}} \left(-\Delta\tilde{\alpha}(x) - \Delta\tilde{\beta}(y) - \Delta\varepsilon \cdot \log(\mu(x)\nu(y)) \right) - \varepsilon_2 \cdot \log(|X_i||Y_j|) \\ &\geq -4|Y_i|L - \varepsilon_2 \cdot \log(|X_i||Y_j|) \end{aligned}$$

where we used $|\Delta\varepsilon \log(\mu(x)\nu(y))| \leq 2\varepsilon_1 \log M = L$. From this follows that $w(i, j) \leq 8 \max\{|Y_i|, |Y_j|\} \cdot \varepsilon_1 \log M + 2\varepsilon_1 \log N$, which in turn implies that $\max \operatorname{diam}(w) \leq 16\varepsilon_1 N \log M + 2\varepsilon_1 R \log N$.

Recall that $\Delta\hat{\beta} = \hat{\beta}_2^\dagger - \hat{\beta}_1^\dagger$, where $\hat{\beta}_a^\dagger$, $a = 1, 2$, are the optimizers of the effective diagonal problems. Then from Lemma 23, and by bounding $R \leq N$ we obtain that

$$\max \Delta\hat{\beta} - \min \Delta\hat{\beta} \leq \varepsilon_1 N (4 \log N + 16 \log M)$$

and finally with $\max \Delta\beta - \min \Delta\beta \leq \max \Delta\tilde{\beta} - \min \Delta\tilde{\beta} + \max \Delta\hat{\beta} - \min \Delta\hat{\beta}$ we get

$$\max \Delta\beta - \min \Delta\beta \leq \varepsilon_1 N (4 \log N + 24 \log M),$$

and analogously we get the equivalent bound for $\Delta\alpha$.

REFERENCES

- [1] M. AGUEH AND G. CARLIER, *Barycenters in the Wasserstein space*, SIAM J. Math. Anal., 43 (2011), pp. 904–924.
- [2] R. K. AHUJA, T. L. MAGNANTI, AND J. B. ORLIN., *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Inc., 1993.
- [3] L. AMBROSIO, N. GIGLI, AND G. SAVARÉ, *Gradient Flows in Metric Spaces and in the Space of Probability Measures*, Lectures in Mathematics, Birkhäuser Boston, 2005.
- [4] H. H. BAUSCHKE AND P. L. COMBETTES, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, CMS Books in Mathematics, Springer, 1st ed., 2011.
- [5] J.-D. BENAMOU AND Y. BRENIER, *A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem*, Numerische Mathematik, 84 (2000), pp. 375–393.
- [6] J.-D. BENAMOU, G. CARLIER, M. CUTURI, L. NENNA, AND G. PEYRÉ, *Iterative Bregman projections for regularized transportation problems*, SIAM J. Sci. Comput., 37 (2015), pp. A1111–A1138, <https://hal.archives-ouvertes.fr/hal-01096124>.
- [7] J.-D. BENAMOU, F. COLLINO, AND J.-M. MIREBEAU, *Monotone and consistent discretization of the Monge–Ampère operator*. arxiv:1409.6694.
- [8] J.-D. BENAMOU, B. D. FROESE, AND A. M. OBERMAN, *Numerical solution of the optimal transportation problem using the Monge–Ampère equation*, Journal of Computational Physics, 260 (2014), pp. 107–126.
- [9] D. P. BERTSEKAS, *The auction algorithm: A distributed relaxation method for the assignment problem*, Annals of Operations Research, 14 (1988), pp. 105–123.
- [10] D. P. BERTSEKAS AND J. ECKSTEIN, *Dual coordinate step methods for linear network flow problems*, Mathematical Programming, Series B, 42 (1988), pp. 203–243.
- [11] Y. BRENIER, *Polar factorization and monotone rearrangement of vector-valued functions*, Comm. Pure Appl. Math., 44 (1991), pp. 375–417.
- [12] G. CARLIER, V. DUVAL, G. PEYRÉ, AND B. SCHMITZER, *Convergence of entropic schemes for optimal transport and gradient flows*, SIAM J. Math. Anal., 49 (2017), pp. 1385–1418.

- [13] L. CHIZAT, G. PEYRÉ, B. SCHMITZER, AND F.-X. VIALARD, *An interpolating distance between optimal transport and Fisher–Rao metrics*, *Found. Comp. Math.*, (2016).
- [14] L. CHIZAT, G. PEYRÉ, B. SCHMITZER, AND F.-X. VIALARD, *Scaling algorithms for unbalanced optimal transport problems*, *Math. Comp.*, 87 (2018), pp. 2563–2609, <https://doi.org/10.1090/mcom/3303>.
- [15] L. CHIZAT, G. PEYRÉ, B. SCHMITZER, AND F.-X. VIALARD, *Unbalanced optimal transport: Dynamic and Kantorovich formulations*, *J. Funct. Anal.*, 27 (2018), pp. 3090–3123, <https://doi.org/10.1016/j.jfa.2018.03.008>.
- [16] R. COMINETTI AND J. SAN MARTIN, *Asymptotic analysis of the exponential penalty trajectory in linear programming*, *Mathematical Programming*, 67 (1992), pp. 169–187.
- [17] M. CUTURI, *Sinkhorn distances: Lightspeed computation of optimal transportation distances*, in *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, 2013, pp. 2292–2300.
- [18] M. CUTURI AND D. AVIS, *Ground metric learning*, *Journal of Machine Learning Research*, 15 (2014), pp. 533–564.
- [19] M. CUTURI AND A. DOUCET, *Fast computation of Wasserstein barycenters*, in *International Conference on Machine Learning*, 2014.
- [20] J. H. FITSCHEN, F. LAUS, AND G. STEIDL, *Transport between RGB images motivated by dynamic optimal transport*, *J. Math. Imaging Vis.*, (2016).
- [21] J. FRANKLIN AND J. LORENZ, *On the scaling of multidimensional matrices*, *Linear Algebra and its Applications*, 114–115 (1989), pp. 717–735.
- [22] A. V. GOLDBERG AND R. E. TARJAN, *Finding minimum-cost circulations by successive approximation*, *Math. Oper. Res.*, 15 (1990), pp. 430–466.
- [23] S. HAKER, L. ZHU, A. TANNENBAUM, AND S. ANGENENT, *Optimal mass transport for registration and warping*, *Int. J. Comp. Vision*, 60 (2004), pp. 225–240.
- [24] R. JORDAN, D. KINDERLEHRER, AND F. OTTO, *The variational formulation of the Fokker-Planck equation*, *SIAM J. Math. Anal.*, 29 (1998), pp. 1–17.
- [25] P. A. KNIGHT, *The Sinkhorn-Knopp algorithm: Convergence and applications*, *SIAM J. Matrix Anal. & Appl.*, 30 (2008), pp. 261–275.
- [26] S. KONDRATYEV, L. MONSANGEON, AND D. VOROTNIKOV, *A new optimal transport distance on the space of finite Radon measures*, *Adv. Differential Equations*, 21 (2016), pp. 1117–1164.
- [27] J. KOSOWSKY AND A. YUILLE, *The invisible hand algorithm: Solving the assignment problem with statistical physics*, *Neural Networks*, 7 (1994), pp. 477–490.
- [28] H. W. KUHN, *The Hungarian method for the assignment problem*, *Naval Research Logistics*, 2 (1955), pp. 83–97.
- [29] C. LÉONARD, *From the Schrödinger problem to the Monge–Kantorovich problem*, *Journal of Functional Analysis*, 262 (2012), pp. 1879–1920.
- [30] B. LÉVY, *A numerical algorithm for L_2 semi-discrete optimal transport in 3D*, *ESAIM Math. Model. Numer. Anal.*, 49 (2015), pp. 1693–1715.
- [31] M. LIERO, A. MIELKE, AND G. SAVARÉ, *Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures*. arxiv:1508.07941, 2015.
- [32] J. MAAS, M. RUMPF, C. SCHÖNLIEB, AND S. SIMON, *A generalized model for optimal transport of images including dissipation and density modulation*, *ESAIM Math. Model. Numer. Anal.*, 49 (2015), pp. 1745–1769.
- [33] M. MANDAD, D. COHEN-STEINER, L. KOBELT, P. ALLIEZ, AND M. DESBRUN, *Variance-minimizing transport plans for inter-surface mapping*. <https://hal.inria.fr/hal-01519006/>, 2017.
- [34] Q. MÉRIGOT, *A multiscale approach to optimal transport*, *Computer Graphics Forum*, 30 (2011), pp. 1583–1592.
- [35] A. M. OBERMAN AND Y. RUAN, *An efficient linear programming method for optimal transportation*. arxiv:1509.03668, 2015.
- [36] G. PEYRÉ, *Entropic approximation of Wasserstein gradient flows*, *SIAM J. Imaging Sci.*, 8 (2015), pp. 2323–2351.
- [37] J. RABIN AND N. PAPADAKIS, *Convex color image segmentation with optimal transport distances*, in *Scale Space and Variational Methods (SSVM 2015)*, 2015, pp. 256–268.
- [38] Y. RUBNER, C. TOMASI, AND L. J. GUIBAS, *The earth mover’s distance as a metric for image retrieval*, *Int. J. Comp. Vision*, 40 (2000), pp. 99–121.
- [39] F. SANTAMBROGIO, *Optimal Transport for Applied Mathematicians*, vol. 87 of *Progress in Nonlinear Differential Equations and Their Applications*, Birkhäuser Boston, 2015.
- [40] B. SCHMITZER, *A sparse multi-scale algorithm for dense optimal transport*, *J. Math. Imaging Vis.*, 56 (2016), pp. 238–259.
- [41] B. SCHMITZER AND C. SCHNÖRR, *A hierarchical approach to optimal transport*, in *Scale Space and Variational Methods (SSVM 2013)*, 2013, pp. 452–464.
- [42] B. SCHMITZER AND C. SCHNÖRR, *Globally optimal joint image segmentation and shape matching based on Wasserstein modes*, *J. Math. Imaging Vis.*, 52 (2015), pp. 436–458, <https://doi.org/10.1007/s10851-014-0546-8>.
- [43] M. SHARIFY, S. GAUBERT, AND L. GRIGORI, *Solution of the optimal assignment problem by diagonal scaling algorithms*. arxiv:1104.3830v2, 2013.
- [44] R. D. SINKHORN AND P. J. KNOPP, *Concerning nonnegative matrices and doubly stochastic matrices*, *Pacific J. Math*, 21 (1967), pp. 343–348.
- [45] J. SOLOMON, F. DE GOES, G. PEYRÉ, M. CUTURI, A. BUTSCHER, A. NGUYEN, T. DU, AND L. GUIBAS, *Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains*, *ACM Transactions on Graphics (Proc. of SIGGRAPH 2015)*, 34 (2015), pp. 66:1–66:11, <http://hal.archives-ouvertes.fr/hal-01188953>.
- [46] M. THORPE, S. PARK, S. KOLOURI, G. K. ROHDE, AND D. SLEPČEV, *A transportation L_p distance for signal analysis*, *J. Math. Imaging Vis.*, (2017), <https://doi.org/10.1007/s10851-017-0726-4>.
- [47] C. VILLANI, *Optimal Transport: Old and New*, vol. 338 of *Grundlehren der mathematischen Wissenschaften*, Springer, 2009.