Comparative Analysis of Computational Methods for Optimal Transport

Ivan Zhytkevych

Thesis submitted to the University of Ottawa in partial fulfillment of the requirements for the degree of Master of Science in Mathematics and Statistics

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

Under the supervision of Mahmoud Zarepour

Abstract

This thesis presents a comparative and experimental study of computational methods for discrete optimal transport under the squared-Euclidean cost. We benchmark the exact simplex method, the Sinkhorn algorithm, first-order and quasi-Newton dual ascent, a chi-square-regularized primal–dual hybrid gradient (PDHG) method, and the Back-and-Forth method, across dimensions $d \in \{1, 2, 3\}$, a range of grid resolutions and regularization levels, and a family of distributions varying in modality, tail thickness, skewness, and anisotropy. Solvers are assessed against three criteria — stability, runtime, and accuracy — and the conclusions are validated on a colour-transfer application. There we find that in RGB colour space the exact solver is not always preferable to a smoother, slightly biased map, which scores better on the structure-preservation metrics of SSIM, gradient correlation, and Laplacian sharpness. Beyond the comparison, we introduce a new method for the outer-regularized 2-Wasserstein barycenter based on Sobolev gradient ascent on the dual functionals combined with the Back-and-Forth c-transform, the simplest case of a broader Back-and-Forth Flow framework.

Acknowledgements

I thank Prof. Augusto Gerolin, who supervised this work, and Prof. Mahmoud Zarepour, who took over as supervisor and saw the thesis through to its defence. I thank the members of my examining committee, Prof. Mohamedou Ould Haye and Prof. François-Michel Boire, for their careful reading and their detailed comments.

I thank Denys Ruban for his companionship in research and for our joint work on the Back-and-Forth Flow framework.

I thank Prof. Tom Woo and the Woo Lab for providing access to their computational cluster, on which the experiments reported in this thesis were run, and Dr. Mohammad Ali Ahmadpoor Jadehkenary for discussions on the functional-analytic aspects of the Back-and-Forth method.

Finally, I sincerely thank my parents, whose love and support will outlast every result and achievement contained in this work.

Author Contributions

This thesis combines a review of established optimal-transport theory and algorithms with the two original contributions described below.

1. **The comparative benchmarking study of optimal-transport solvers** (Sections 4.1 to 4.3 and 4.5), which is entirely my own work. It comprises:
   (a) the `uot-bench` benchmarking framework and the solver implementations it contains;
   (b) the implementation of the Back-and-Forth method for arbitrary dimension $d \geq 1$, including the CIC and adaptive pushforward schemes (Appendices B.2 and B.3);
   (c) the derivation of the discretization floor of the Monge–Ampère residual (Appendix A.4), which is what makes the residual usable as a comparative measure of structural accuracy;
   (d) the design and execution of the experiments across dimensions $d \in \{1, 2, 3\}$, grid resolutions, regularization levels and distribution families, and the stability, runtime and accuracy analysis drawn from them;
   (e) the colour-transfer study (Section 4.5), including its evaluation protocol and metric analysis.
2. **The outer-regularized Back-and-Forth method for 2-Wasserstein barycenters** (Section 4.4), the simplest case of the more general Back-and-Forth Flow (BaFFO) framework developed jointly with Denys Ruban and Prof. Augusto Gerolin; a manuscript based on that framework is currently under review (Zhytkevych et al., 2026). The theory presented in this section — the duality result (Theorem 4.4.2) and the first variation of the dual functionals (Proposition 4.4.4) — was developed jointly with Denys Ruban. The implementation of the method for 2-Wasserstein barycenters and the numerical study reported here are my own work. The general framework of Section 4.4.3 is likewise joint work with Denys Ruban: I formulated the framework on a graph of measures, Ruban developed the theory for its applications to mean-field games and trajectory inference, and the implementations were carried out together. The duality and first-variation derivations are given at a formal level; making them rigorous is part of the ongoing collaborative work.

Contents

List of Figures

List of Tables

List of Abbreviations

BaFFO	Back-and-Forth Flow
BFS	basic feasible solution
CIC	Cloud-in-Cell
CIELAB	CIE $L^*a^*b^*$ colour space
DCT	discrete cosine transform
FFT	fast Fourier transform
GH	generalized hyperbolic
GMM	Gaussian mixture model
GPU	graphics processing unit
IQR	interquartile range
KL	Kullback–Leibler
L-BFGS	limited-memory Broyden–Fletcher–Goldfarb–Shanno
LP	linear program
LSE	log-sum-exp
MFG	mean-field game
OLS	ordinary least squares
OT	optimal transport
PDHG	primal–dual hybrid gradient
POT	Python Optimal Transport (library)
RGB	red–green–blue colour space
SGD	stochastic gradient descent
SSIM	structural similarity index
TV	total variation
XLA	accelerated linear algebra

Notation and Conventions

The following symbols and conventions are used consistently throughout the thesis.

Definitions. The symbol $:=$ marks a definition: the object on its left is introduced by the expression on its right. A plain $=$ states an identity between objects that are already defined.

Star convention. For every variational objective F, the starred symbol $F^\star$ denotes its *optimal value* (infimum or supremum); the unstarred symbol F always denotes the *objective functional* evaluated at a specific argument. For example, $\mathcal{K}(\pi)$ is the transport cost of a given plan π, while $\mathcal{K}^\star(\mu,\nu) := \inf_{\pi\in\Pi(\mu,\nu)} \mathcal{K}(\pi)$ is its infimum.

Spaces and measures.

$d \geq 1$ Dimension of the ambient space $\mathbb{R}^d$.

$\Omega \subseteq \mathbb{R}^d$ Compact domain of the transport problem. Convexity of Ω is imposed only where explicitly stated.

$\mathcal{B}(\Omega)$ Borel σ-algebra of Ω.

$\mathcal{M}^+(\Omega)$ Space of non-negative Borel measures on Ω.

$\mathcal{P}(\Omega)$ Space of Borel probability measures on Ω: $\mathcal{P}(\Omega) := \{\mu \in \mathcal{M}^+(\Omega) : \mu(\Omega) = 1\}$.

δ_{x_0} Dirac mass at $x_0 \in \Omega$: $\delta_{x_0}(A) := \mathbf{1}\{x_0 \in A\}$ for $A \in \mathcal{B}(\Omega)$. Not to be confused with the variation $\delta F(\varphi)[h]$ below.

$\mathcal{P}_p(\mathbb{R}^d)$ Wasserstein space of probability measures on $\mathbb{R}^d$ with finite p-th moment: $\{\mu \in \mathcal{P}(\mathbb{R}^d) : \int \|x\|^p \,\mathrm{d}\mu < \infty\}$.

μ, ν Source and target probability measures in $\mathcal{P}(\Omega)$, respectively.

$C(\Omega)$ Space of real-valued continuous functions on Ω.

$C_b(\Omega)$, $C_b(\Omega \times \Omega)$ Spaces of bounded continuous real-valued functions on Ω and on $\Omega \times \Omega$; the cost c lives in the latter.

$\mathcal{L}^d$ d-dimensional Lebesgue measure.

Transport plans, maps, and objectives.

$T_\# \mu$ Pushforward of μ under a Borel map $T : \Omega \to \Omega$: $T_\#\mu(A) := \mu(T^{-1}(A))$ for every $A \in \mathcal{B}(\Omega)$.

p_1, p_2 Coordinate projections on $\Omega \times \Omega$: $\mathrm{p}_1(x,y) := x$, $\mathrm{p}_2(x,y) := y$.

$\Pi(\mu,\nu)$ Set of admissible transport plans (couplings): $\Pi(\mu,\nu) := \{\pi \in \mathcal{P}(\Omega \times \Omega) : (\mathrm{p}_1)_\#\pi = \mu,\ (\mathrm{p}_2)_\#\pi = \nu\}$.

$c(x, y)$ Transport cost function. The default throughout the numerical experiments is the squared Euclidean cost $c(x, y) = \|x - y\|^2$.

$\mathcal{K}(\pi)$ Kantorovich transport cost of a plan π: $\mathcal{K}(\pi) := \int_{\Omega\times\Omega} c(x, y)\,\mathrm{d}\pi(x, y)$.

$\mathcal{K}^\star(\mu, \nu)$ Optimal Kantorovich value: $\mathcal{K}^\star(\mu, \nu) := \inf_{\pi\in\Pi(\mu,\nu)} \mathcal{K}(\pi)$.

$\mathcal{T}(\mu, \nu)$ Set of admissible Monge maps: $\{T : \Omega \to \Omega \text{ Borel} : T_\#\mu = \nu\}$.

$\mathcal{M}(T)$ Monge transport cost of a map T: $\mathcal{M}(T) := \int_\Omega c(x, T(x))\,\mathrm{d}\mu(x)$.

$\mathcal{M}^\star(\mu, \nu)$ Optimal Monge value: $\mathcal{M}^\star(\mu, \nu) := \inf_{T\in\mathcal{T}(\mu,\nu)} \mathcal{M}(T)$, with the convention $\inf \varnothing = +\infty$.

Dual potentials and the c-transform.

φ, ψ Kantorovich dual potentials; by convention φ is the source-side potential and ψ the target-side potential, both in $C(\Omega)$.

$\varphi \oplus \psi$ Lifted sum: $(x, y) \mapsto \varphi(x)+\psi(y)$. The constraint $\varphi\oplus\psi \le c$ abbreviates $\varphi(x)+\psi(y) \le c(x, y)$ for all $(x, y) \in \Omega \times \Omega$.

φ^c c-transform of φ: $\varphi^c(y) := \inf_{x\in\Omega}\{c(x, y) - \varphi(x)\}$.

c-conc(Ω) Set of c-concave functions: $\{\varphi : \Omega \to \mathbb{R} \cup \{+\infty\} : \varphi = \psi^c \text{ for some } \psi\}$.

$\mathcal{D}(\varphi, \psi)$ Kantorovich dual objective: $\mathcal{D}(\varphi, \psi) := \int_\Omega \varphi\,\mathrm{d}\mu + \int_\Omega \psi\,\mathrm{d}\nu$.

$\mathcal{D}^\star(\mu, \nu)$ Optimal dual value: $\mathcal{D}^\star(\mu, \nu) := \sup_{\varphi\oplus\psi\le c} \mathcal{D}(\varphi, \psi)$.

Entropic regularization.

$\varepsilon > 0$ Entropic regularization parameter. Smaller ε brings the regularized problem closer to the unregularized linear program.

$\mathcal{H}(\pi \,\|\, \xi)$ Relative entropy (Kullback–Leibler divergence) of π with respect to ξ, equal to $\int \log(\mathrm{d}\pi/\mathrm{d}\xi)\,\mathrm{d}\pi$ when $\pi \ll \xi$ and $+\infty$ otherwise.

$\mathbb{G}$ Gibbs kernel with density $\mathrm{d}\mathbb{G}(x, y) := e^{-c(x,y)/\varepsilon}\,\mathrm{d}\mu(x)\,\mathrm{d}\nu(y)$ against $\mu \otimes \nu$.

$\mathcal{K}_\varepsilon(\pi)$ Entropic transport objective: $\mathcal{K}_\varepsilon(\pi) := \mathcal{K}(\pi) + \varepsilon\,\mathcal{H}(\pi \,\|\, \mu \otimes \nu)$.

$\mathcal{K}^\star_\varepsilon(\mu, \nu)$ Optimal entropic value: $\mathcal{K}^\star_\varepsilon(\mu, \nu) := \min_{\pi\in\Pi(\mu,\nu)} \mathcal{K}_\varepsilon(\pi)$.

π^ε Unique minimiser of $\mathcal{K}_\varepsilon$ over $\Pi(\mu, \nu)$.

$\mathcal{D}_\varepsilon(\varphi, \psi)$ Entropic dual objective (unconstrained in φ, ψ); see (3.0.5) for the discrete form.

$\mathcal{D}^\star_\varepsilon$ Optimal entropic dual value: $\mathcal{D}^\star_\varepsilon := \sup_{\varphi,\psi} \mathcal{D}_\varepsilon(\varphi, \psi)$.

Wasserstein distance.

$W_p(\mu, \nu)$ p-Wasserstein distance, $p \ge 1$: $W_p(\mu, \nu) := \big(\inf_{\pi\in\Pi(\mu,\nu)} \int \|x - y\|^p\,\mathrm{d}\pi\big)^{1/p}$. The default in the experiments is $p = 2$.

Sobolev spaces and the Back-and-Forth method.

$\dot{H}^1(\Omega)$ Homogeneous Sobolev space of zero-mean functions with square-integrable gradient: $\{\varphi : \int_\Omega \varphi \,\mathrm{d}x = 0,\ \nabla\varphi \in L^2(\Omega)\}$, with inner product $\langle \varphi_1, \varphi_2 \rangle_{\dot{H}^1} := \int_\Omega \nabla\varphi_1 \cdot \nabla\varphi_2 \,\mathrm{d}x$.

$\dot{H}^{-1}(\Omega)$ Dual of $\dot{H}^1(\Omega)$.

$H^1(\Omega)$ Inhomogeneous Sobolev space $\{u \in L^2(\Omega) : \nabla u \in L^2(\Omega)\}$, with inner product $\langle u, v \rangle_{H^1} := \int_\Omega (uv + \nabla u \cdot \nabla v) \,\mathrm{d}x$; used in the barycenter section.

$\langle \cdot, \cdot \rangle_{\dot{H}^1}$ Inner product on $\dot{H}^1(\Omega)$: $\langle \varphi_1, \varphi_2 \rangle_{\dot{H}^1} := \int_\Omega \nabla\varphi_1(x) \cdot \nabla\varphi_2(x) \,\mathrm{d}x$.

$\langle \cdot, \cdot \rangle_{H^1}$ Inner product on $H^1(\Omega)$ (used in the barycenter section): $\langle u, v \rangle_{H^1} := \int_\Omega \big(u\, v + \nabla u \cdot \nabla v\big) \,\mathrm{d}x$.

$J(\varphi)$, $I(\psi)$ Back-and-Forth dual functionals: $J(\varphi) := \int_\Omega \varphi \,\mathrm{d}\mu + \int_\Omega \varphi^c \,\mathrm{d}\nu$; $\quad I(\psi) := \int_\Omega \psi \,\mathrm{d}\nu + \int_\Omega \psi^c \,\mathrm{d}\mu$.

$\delta F(\varphi)[h]$ Gâteaux (first-variation) derivative of F at φ in direction h.

$\nabla_{\dot{H}^1} J(\varphi)$ $\dot{H}^1$-gradient of J: the unique Riesz representative of $\delta J(\varphi)$ in $\dot{H}^1(\Omega)$; equals $(-\Delta)^{-1}(\mu - (S_\varphi)_\# \nu)$.

$(-\Delta)^{-1}\rho$ Zero-mean solution of the Neumann–Poisson problem $-\Delta g = \rho$ on Ω, $\partial_n g = 0$ on $\partial\Omega$, $\int_\Omega g \,\mathrm{d}x = 0$; computed spectrally via DCT at cost $O(N \log N)$, $N = n^d$.

σ_k Step size at iteration k of the Back-and-Forth scheme; chosen by Armijo–Goldstein line search.

$\mathrm{smin}_\varepsilon\{z_j\}$ Soft (entropic) minimum: $-\varepsilon \ln \sum_j \exp(-z_j/\varepsilon) \to \min_j z_j$ as $\varepsilon \to 0^+$.

Wasserstein barycenter and outer regularization.

$\gamma > 0$ Outer regularization parameter in the barycenter problem; distinct from the entropic parameter ε.

$\lambda_1, \ldots, \lambda_N$ Barycenter weights: $\lambda_i \geq 0$, $\sum_{i=1}^N \lambda_i = 1$.

$\nu^\star$, $\nu_\gamma^\star$ Wasserstein barycenter and its outer-regularized counterpart.

$\mathcal{R}(\nu)$ Entropy functional: $\mathcal{R}(\nu) := \int_\Omega \nu(y)(\ln \nu(y) - 1) \,\mathrm{d}y$ for absolutely continuous ν, $+\infty$ otherwise.

$\mathrm{Bar}_\gamma(\nu; \mu_1, \ldots, \mu_N)$ Regularized barycenter objective: $\sum_{i=1}^N \lambda_i W_2^2(\mu_i, \nu) + \gamma \mathcal{R}(\nu)$.

Discrete setting.

n, m Numbers of support points of the discrete source and target, respectively.

$\mathcal{X} = \{x_i\}_{i=1}^n$, $\mathcal{Y} = \{y_j\}_{j=1}^m$ Discrete support sets of the source and target.

$a \in \mathbb{R}_+^n$, $b \in \mathbb{R}_+^m$ Probability weight vectors: $a_i, b_j \geq 0$, $\sum_i a_i = \sum_j b_j = 1$.

$\pi \in \mathbb{R}_{0,+}^{n \times m}$ Discrete transport matrix (plan).

$C \in \mathbb{R}^{n \times m}$ Cost matrix: $C_{ij} := c(x_i, y_j)$.

$K \in \mathbb{R}^{n \times m}$ Gibbs kernel matrix: $K_{ij} := \exp(-C_{ij}/\varepsilon)$.

$[n]$ Index set $\{1, 2, \ldots, n\}$.

$\mathbf{1}_m$ Column vector of m ones; $\mathbf{1}_m^\top$ denotes its transpose.

I_n $n \times n$ identity matrix.

$A_{[B]}$ Submatrix of A formed by the columns indexed by $B \subseteq \mathbb{N}$.

$\mathrm{vec}(\pi)$ Row-major vectorisation of $\pi \in \mathbb{R}^{n \times m}$ into a vector in $\mathbb{R}^{nm}$.

Operators and shorthand.

$f \oplus g$ For $f : \mathcal{X} \to \mathbb{R}$ and $g : \mathcal{Y} \to \mathbb{R}$: $(f \oplus g)(x, y) := f(x) + g(y)$. Used for both continuous potentials and their discrete counterparts.

$\mu \otimes \nu$ Product measure of μ and ν; $\otimes$ also denotes the Kronecker product when applied to matrices.

$\langle C, \pi \rangle$ Frobenius inner product of two matrices of the same size: $\langle C, \pi \rangle := \sum_{i,j} C_{ij} \pi_{ij}$. it denotes the corresponding Hilbert-space inner product.

T_π Barycentric projection of a transport plan π: $T_\pi(x_i) := a_i^{-1} \sum_j \pi_{ij} y_j$. Used to extract a Monge-type map from a Kantorovich plan.

T_α Displacement interpolation at level $\alpha \in [0, 1]$: $T_\alpha(x) := (1 - \alpha)x + \alpha T(x)$. Used in the colour-transfer experiments.

Part 1

Introduction and Preliminaries

Optimal transport has become a standard computational tool, but the choice of algorithm used to compute it is rarely examined: solvers are usually adopted from a library and compared, if at all, only against the one they were designed to replace. This chapter states the question the thesis sets out to answer and summarizes what it contributes.

1.1 Motivation

Optimal transport (OT) is, in its Kantorovich form, an infinite-dimensional linear program (LP) (Santambrogio, 2015). Its discretization on a grid of n points turns it into an LP with n^2 variables (Peyré and Cuturi, 2019, Chapter 3), whose exact solution by simplex or interior-point methods scales poorly and becomes impractical in higher dimensions (Altschuler et al., 2017). The entropic regularization approach popularized by Cuturi (2013), solved by the Sinkhorn algorithm, changed this. By turning the problem into a sequence of cheap matrix–vector products well suited to graphics processing units (GPUs), it became the default tool across computer vision (Solomon et al., 2015), statistics (Panaretos and Zemel, 2020), economics (Galichon, 2016) and machine learning applications (Arjovsky et al., 2017).

The robustness of the method comes at a price: the entropic term in the objective implies a bias that blurs the transport plan and does not vanish under grid refinement at fixed regularization. A range of alternative methods were developed to avoid this, among them the Back-and-Forth method of Jacobs and Léger (2020) and primal–dual splitting such as the primal–dual hybrid gradient (PDHG) method of Ruban and Gerolin (2025). Each makes a different trade-off between accuracy, stability and cost. Yet these solvers are rarely compared on a common footing.

Benchmarking is not new to optimal transport, but the existing studies cover a narrow slice of the solver landscape. The closest precedent is DOTmark (Schrieber et al., 2017): a neutral collection of two-dimensional image histograms, used to compare transportation simplex, shielding and power-diagram methods against general-purpose LP solvers. Its conclusion is that no single method dominates across instance types and resolutions. Only exact solvers of the linear-programming family are considered, however, and neither entropic nor map-based methods appear. Dong et al. (2020) go a step further and compare combinatorial solvers against matrix-scaling ones, on point clouds, document embeddings and image pixels. The combinatorial methods come out faster at comparable accuracy. Their instances are not grids, though, and the regularization parameter is never varied. Their experiments are also single-threaded and run on CPU, and GPU parallelization is not discussed. That bears directly on the conclusion: matrix-scaling methods are the ones a GPU accelerates, whereas network-simplex-type solvers are not readily parallelized. Benchmarks built on problems with analytically known solutions do exist, but they cover only neural solvers in high dimension (Korotin et al., 2021).

The surveys are of little help here. They compare algorithms through complexity bounds rather than through runtime and accuracy measured on shared instances (Peyré and Cuturi, 2019; Mérigot and Thibert, 2021), and the most recent one is no exception (Moradi, 2025). The libraries do not fill the gap either: the reference implementation of the field reports no performance comparison at all, only a table of which algorithms each package provides (Flamary et al., 2021). The Back-and-Forth method is the clearest case. The paper that introduces it (Jacobs and Léger, 2020) validates it on examples with known solutions and dismisses the alternatives in prose, without a single head-to-head experiment. We are not aware of a study that places exact linear programming, entropic, dual first-order, primal–dual and map-based solvers on a common testbed. Nor of one that varies the ambient dimension, the grid resolution, the regularization level and the distribution family together. That is the gap this thesis addresses.

This thesis therefore asks a single question: across dimensions, grid resolutions, regularization levels and distribution families, how do the main families of OT solvers trade accuracy against runtime and stability, and under what conditions is the exact solver not the one to choose? This thesis is motivated by that gap: a controlled, reproducible comparison of the main computational approaches to OT, and, growing out of it, a new method for the regularized Wasserstein barycenter problem.

1.2 Main contributions

This thesis makes three contributions.

The first is an *open, reproducible benchmarking framework*, `uot-bench`, implemented in JAX with GPU support and configuration-driven experiments[1], which is the main software artifact of this work. Two of the solvers are implemented here rather than taken from a library. The reference implementation released with the Back-and-Forth method is two-dimensional and runs on CPU; the one used here covers arbitrary dimension $d \geq 1$ and runs on GPU. Log-domain Sinkhorn is implemented in the same framework, so the entropic and the map-based solvers are exercised through one code path and one hardware setup.

The second is a *systematic comparison of the principal families of OT solvers* — log-domain Sinkhorn, exact simplex, first-order and L-BFGS dual methods (Genevay et al., 2016), chi-square-regularized PDHG (Ruban and Gerolin, 2025), and the Back-and-Forth method (Jacobs and Léger, 2020) — assessed along stability, runtime, and accuracy across dimensions $d \in \{1, 2, 3\}$, grid resolutions, regularization levels, and a range of distribution families, and validated on a real application (colour transfer).

The third is a *new computational method for the outer-regularized* 2*-Wasserstein barycenter*. It carries the Back-and-Forth scheme over from a pair of measures to N marginals, with the regularization acting on the barycenter rather than on the transport plans. The method is the simplest instance of a more general framework, *Back-and-Forth Flow* (BaFFO), developed with Denys Ruban and Prof. Augusto Gerolin for transportation problems on a graph of measures (Zhytkevych et al., 2026). Within it, the formulation on a graph of measures is my own, while the duality and first-variation derivations are joint work with Denys Ruban and are carried out at a formal level. The implementation and the analysis reported here are mine. The Author Contributions statement sets out the division of work in full.

1.3 Outline of the thesis

Part 2 recalls the mathematical background on which the rest of the thesis rests: the Kantorovich and Monge problems, Kantorovich duality and the c-transform, Brenier's theorem, and the Wasserstein distances. Part 3 then derives the solvers that are compared, each from its own formulation. It also fixes what each solver returns, a coupling or a map, which makes some later comparisons possible and others meaningless. The numerical study follows in Part 4. It opens with the experimental protocol and the distribution families, then reports stability, runtime and accuracy on the synthetic benchmark. Colour transfer then tests whether the ranking obtained on synthetic instances survives on a real task. Section 4.4 introduces the outer-regularized barycenter method. Part 5 collects the findings, the limitations, and directions for future work.

[1] `https://github.com/AI-Quantum-Research-Group/uot-bench`

Part 2

Optimal Transport

This part states the transport problem in the forms that the rest of the thesis computes. The Kantorovich problem comes first, and its unknown is a transport plan. The Monge problem asks instead for a map, and the dual problem for a pair of potentials. The Kantorovich problem and its dual always have the same value. The Monge problem can have a larger one, and sometimes no solution at all. The c-transform then cuts the dual down to a single potential. Entropic regularization makes the problem strictly convex, which buys a unique solution and a cheap iteration at the price of a bias. Each solver in Part 3 works on one of these objects, and that is what decides what the solver returns and how it can be compared in Part 4.

Symbols follow the list in Notation and Conventions (p. xi): there $\mathcal{P}(\Omega)$ is the set of Borel probability measures on Ω, and $C_b(\Omega \times \Omega)$ the space of bounded continuous real-valued functions on $\Omega \times \Omega$. Let $\Omega \subset \mathbb{R}^d$ be a compact set, let $\mu, \nu \in \mathcal{P}(\Omega)$, and let $c \in C_b(\Omega \times \Omega)$ be the cost function.

2.1 Kantorovich primal problem

Definition 2.1.1 (Pushforward (Santambrogio, 2015))**.** Let $T : \Omega \to \Omega$ be Borel measurable and $\mu \in \mathcal{P}(\Omega)$. The *pushforward* $T_{\#}\mu \in \mathcal{P}(\Omega)$ is $T_{\#}\mu(A) := \mu\big(T^{-1}(A)\big)$ for every Borel $A \subseteq \Omega$, equivalently $T_{\#}\mu = \mu \circ T^{-1}$.

The corresponding integral characterization of the pushforward is standard (Santambrogio, 2015, Section 1.1). Specifically, $T_{\#}\mu$ is the unique measure with

$$\int_\Omega f \,\mathrm{d}(T_{\#}\mu) = \int_\Omega f \circ T \,\mathrm{d}\mu \qquad \text{for all bounded Borel } f.$$

Example 2.1.2 (Dirac mass)**.** For the Dirac mass δ_{x_0}, defined by $\delta_{x_0}(A) := \mathbf{1}\{x_0 \in A\}$, the pushforward is $T_{\#}\delta_{x_0} = \delta_{T(x_0)}$: all the mass at x_0 is moved to $T(x_0)$. Indeed

$$T_{\#}\delta_{x_0}(A) = \delta_{x_0}(T^{-1}(A)) = \mathbf{1}\{x_0 \in T^{-1}(A)\} = \mathbf{1}\{T(x_0) \in A\} = \delta_{T(x_0)}(A).$$

Since the pushforward is additive, a discrete measure $\mu = \sum_{i=1}^n a_i \delta_{x_i}$ pushes to $T_{\#}\mu = \sum_{i=1}^n a_i \delta_{T(x_i)}$. The weights stay put and only the support points move.

Definition 2.1.3 (Transport plans)**.** A *transport plan*, or *coupling*, between μ and ν is a probability measure on $\Omega \times \Omega$ with first marginal μ and second marginal ν. The set of all such plans is

$$\Pi(\mu, \nu) := \{\pi \in \mathcal{P}(\Omega \times \Omega) : (\mathrm{p}_1)_{\#}\pi = \mu,\ (\mathrm{p}_2)_{\#}\pi = \nu\},$$

where $\mathrm{p}_1(x, y) := x$ and $\mathrm{p}_2(x, y) := y$ are the coordinate projections.

The two constraints read $\pi(A \times \Omega) = \mu(A)$ and $\pi(\Omega \times B) = \nu(B)$ for all Borel $A, B \subseteq \Omega$: all of μ leaves and all of ν arrives. The number $\pi(A \times B)$ is then the mass taken from A and delivered to B. Unlike a map, a plan may split the mass sitting at a point between several destinations. The set $\Pi(\mu, \nu)$ is never empty, since the product measure $\mu \otimes \nu$ always belongs to it.

Definition 2.1.4 (Kantorovich primal problem)**.** The transport cost of a plan $\pi \in \Pi(\mu, \nu)$ and its optimal value are, respectively,

$$\mathcal{K}(\pi) := \int_{\Omega \times \Omega} c(x, y) \,\mathrm{d}\pi(x, y), \qquad \mathcal{K}^\star(\mu, \nu) := \inf_{\pi \in \Pi(\mu,\nu)} \mathcal{K}(\pi). \tag{2.1.1}$$

Here the infimum defining $\mathcal{K}^\star$ in (2.1.1) is taken over the set $\Pi(\mu, \nu)$ of admissible transport plans, that is, the probability measures $\pi \in \mathcal{P}(\Omega \times \Omega)$ whose first and second marginals are μ and ν,

respectively. The existence of a minimizer (which we will call *the optimal transport plan*) for this problem is guaranteed under our assumptions. We refer to Santambrogio (2015, Theorem 1.4) for a complete proof.

Remark 2.1.5 (Probabilistic formulation (Peyré and Cuturi, 2019)). The Kantorovich problem admits a probabilistic reading. Any coupling $\pi \in \Pi(\mu,\nu)$ is the joint law of a pair of random variables (X,Y) taking values in $\Omega \times \Omega$ whose marginals are $X \sim \mu$ and $Y \sim \nu$, and conversely every such pair induces a coupling $\pi = \mathrm{Law}(X,Y)$, where $\mathrm{Law}(X,Y)(A) := \mathbb{P}((X,Y) \in A)$ for every Borel $A \subseteq \Omega \times \Omega$. Under this identification the transport cost $\mathcal{K}(\pi)$ of (2.1.1) is the expected cost $\mathbb{E}_{(X,Y)\sim\pi}\big[c(X,Y)\big]$, so the Kantorovich problem becomes the minimization of this expectation over all admissible laws π,

$$\mathcal{K}^\star(\mu,\nu) = \inf_{\pi\in\Pi(\mu,\nu)} \mathbb{E}_{(X,Y)\sim\pi}\big[c(X,Y)\big].$$

2.2 Dual problem

The dual problem arises from the primal by *Lagrangian duality*; we give the derivation formally, the rigorous identity being Theorem 2.2.3.[2] Relax the two marginal constraints defining $\Pi(\mu,\nu)$ by introducing multipliers — the potentials $\varphi, \psi \in C(\Omega)$ — and form the Lagrangian, for nonnegative measures $\pi \geq 0$ on $\Omega \times \Omega$,

$$\begin{aligned}\mathcal{L}(\pi,\varphi,\psi) := &\int_{\Omega\times\Omega} c(x,y)\,\mathrm{d}\pi(x,y)\\ &+ \Big(\int_\Omega \varphi(x)\,\mathrm{d}\mu(x) - \int_{\Omega\times\Omega} \varphi(x)\,\mathrm{d}\pi(x,y)\Big)\\ &+ \Big(\int_\Omega \psi(y)\,\mathrm{d}\nu(y) - \int_{\Omega\times\Omega} \psi(y)\,\mathrm{d}\pi(x,y)\Big).\end{aligned}$$

Maximizing over φ, ψ enforces the marginal constraints. If $(\mathrm{p}_1)_\#\pi \neq \mu$, the first parenthesis is a nonzero linear functional of φ, so its supremum over φ is $+\infty$, and the second parenthesis behaves the same way in ψ when $(\mathrm{p}_2)_\#\pi \neq \nu$. If both marginals are correct, the two parentheses vanish and $\mathcal{L}(\pi,\varphi,\psi) = \mathcal{K}(\pi)$. Hence the primal value is the inf–sup

$$\mathcal{K}^\star(\mu,\nu) = \inf_{\pi\geq 0} \sup_{\varphi,\psi} \mathcal{L}(\pi,\varphi,\psi).$$

Exchanging the order and regrouping $\mathcal{L}(\pi,\varphi,\psi) = \int_\Omega \varphi\,\mathrm{d}\mu + \int_\Omega \psi\,\mathrm{d}\nu + \int_{\Omega\times\Omega}(c - \varphi\oplus\psi)\,\mathrm{d}\pi$, the inner minimization is

$$\inf_{\pi\geq 0} \int_{\Omega\times\Omega} \big(c - \varphi\oplus\psi\big)\,\mathrm{d}\pi = \begin{cases} 0, & \varphi\oplus\psi \leq c,\\ -\infty, & \text{otherwise.}\end{cases}$$

If $c - \varphi\oplus\psi$ were negative at some point, concentrating ever more mass there would drive the integral to $-\infty$. If it is nonnegative everywhere, the integral is nonnegative for every π and vanishes at $\pi = 0$, so the infimum is attained there. The value $-\infty$ discards exactly the pairs violating $\varphi\oplus\psi \leq c$, so the surviving supremum runs over the admissible potentials and leaves the objective $\int_\Omega \varphi\,\mathrm{d}\mu + \int_\Omega \psi\,\mathrm{d}\nu$. This motivates the following definition.

[2] Only one step below is formal: the exchange of the infimum and the supremum. A minimax theorem can justify such an exchange (Sion, 1958), but we do not verify its hypotheses here. The rigorous identity is stated in Theorem 2.2.3, and Santambrogio (2015, Section 1.6) gives a proof.

Definition 2.2.1 (Kantorovich dual problem)**.** The dual objective and its optimal value (the Kantorovich dual problem associated with (2.1.1)) are, respectively,

$$\mathcal{D}(\varphi,\psi) := \int_\Omega \varphi \,\mathrm{d}\mu + \int_\Omega \psi \,\mathrm{d}\nu, \qquad \mathcal{D}^\star(\mu,\nu) := \sup_{\varphi\oplus\psi\le c} \mathcal{D}(\varphi,\psi), \tag{2.2.1}$$

where $\varphi,\psi \in C(\Omega)$ and $\varphi\oplus\psi \le c$ abbreviates $\varphi(x)+\psi(y)\le c(x,y)$ for all $(x,y)\in\Omega\times\Omega$. A pair (φ,ψ) satisfying this constraint is called *admissible*.

Weak duality states that the dual value can never exceed the primal one. Strong duality states that the two values are equal.

Lemma 2.2.2 (Weak duality (Santambrogio, 2015))**.** *For every $\pi\in\Pi(\mu,\nu)$ and every admissible pair (φ,ψ),*

$$\mathcal{D}(\varphi,\psi) = \int_\Omega \varphi \,\mathrm{d}\mu + \int_\Omega \psi \,\mathrm{d}\nu \;\le\; \int_{\Omega\times\Omega} c(x,y)\,\mathrm{d}\pi(x,y) = \mathcal{K}(\pi). \tag{2.2.2}$$

Taking the supremum over admissible (φ,ψ) and the infimum over $\pi\in\Pi(\mu,\nu)$ yields

$$\mathcal{D}^\star(\mu,\nu) \;\le\; \mathcal{K}^\star(\mu,\nu).$$

Proof. The marginal conditions $(\mathrm{p}_1)_\#\pi=\mu$, $(\mathrm{p}_2)_\#\pi=\nu$ give $\int_{\Omega\times\Omega}\big(\varphi(x)+\psi(y)\big)\,\mathrm{d}\pi = \int_\Omega\varphi\,\mathrm{d}\mu + \int_\Omega\psi\,\mathrm{d}\nu = \mathcal{D}(\varphi,\psi)$. Admissibility $\varphi\oplus\psi\le c$ and $\pi\ge 0$ then give $\mathcal{D}(\varphi,\psi) = \int_{\Omega\times\Omega}(\varphi\oplus\psi)\,\mathrm{d}\pi \le \int_{\Omega\times\Omega} c\,\mathrm{d}\pi = \mathcal{K}(\pi)$, which is (2.2.2). The bound holds for all admissible pairs and all π, so the left side's supremum is $\le$ the right side's infimum. □

Theorem 2.2.3 (Kantorovich strong duality (Santambrogio, 2015, Theorem 1.39))**.** *The primal* (2.1.1) *and dual* (2.2.1) *values coincide,*

$$\mathcal{K}^\star(\mu,\nu) = \mathcal{D}^\star(\mu,\nu).$$

Remark 2.2.4 (Nonuniqueness of dual maximizers)*.* Dual maximizers need not be unique. In particular, if $(\varphi^\star,\psi^\star)$ is a dual maximizing pair, then $(\varphi^\star+a,\psi^\star-a)$ is also dual maximizing for every $a\in\mathbb{R}$. This invariance shows that dual potentials are determined only up to an additive constant. See Santambrogio (2015, Section 1.6) for a discussion.

In linear programming, complementary slackness says that an optimal solution puts weight only where the corresponding constraint is tight. In transport terms, the optimal transport plan lives where the dual constraint holds with equality.

Proposition 2.2.5 (Complementary slackness (Santambrogio, 2015))**.** *Let $\pi^\star$ solve the primal problem in* (2.1.1) *and let $(\varphi^\star,\psi^\star)$ solve the dual problem in* (2.2.1)*. Then*

$$\varphi^\star(x)+\psi^\star(y) = c(x,y) \tag{2.2.3}$$

holds $\pi^\star$-almost everywhere on $\Omega\times\Omega$, and consequently

$$\mathrm{spt}(\pi^\star) \subseteq \big\{(x,y)\in\Omega\times\Omega : \varphi^\star(x)+\psi^\star(y) = c(x,y)\big\}.$$

Proof. The marginal conditions and strong duality (Theorem 2.2.3) give

$$\int_{\Omega\times\Omega}\big(c-\varphi^\star\oplus\psi^\star\big)\,\mathrm{d}\pi^\star = \mathcal{K}(\pi^\star) - \mathcal{D}(\varphi^\star,\psi^\star) = \mathcal{K}^\star(\mu,\nu) - \mathcal{D}^\star(\mu,\nu) = 0.$$

The integrand is nonnegative by admissibility, so it vanishes $\pi^\star$-almost everywhere, which is (2.2.3). Since $c, \varphi^\star, \psi^\star$ are continuous, the set where the equality holds is closed, and by (2.2.3) it has full $\pi^\star$-measure. The support — the smallest closed set of full measure — is therefore contained in that set. □

2.3 Monge problem

The Monge problem asks for a map instead of a plan. A map sends each point to a single destination, so the mass sitting at a point travels together and cannot be divided. Historically this is the original formulation of the transport problem, and the Kantorovich problem of Section 2.1 is its relaxation. The two formulations are compared in Theorem 2.3.2 below.

Definition 2.3.1 (Monge problem)**.** The set of admissible transport maps is

$$\mathcal{T}(\mu,\nu) := \{\, T : \Omega \to \Omega \text{ Borel measurable such that } T_{\#}\mu = \nu \,\}.$$

The Monge cost of a map $T \in \mathcal{T}(\mu,\nu)$ and its optimal value are, respectively,

$$\mathcal{M}(T) := \int_\Omega c\big(x, T(x)\big)\,\mathrm{d}\mu(x), \qquad \mathcal{M}^\star(\mu,\nu) := \inf_{T\in\mathcal{T}(\mu,\nu)} \mathcal{M}(T), \tag{2.3.1}$$

with the convention $\inf \varnothing = +\infty$.

Remark 2.3.2. The Kantorovich formulation extends the Monge formulation by allowing mass splitting. The Monge admissible maps form a nonconvex class and may fail to exist, whereas $\Pi(\mu,\nu)$ is always nonempty. Therefore, the Kantorovich dual problem is associated with the Kantorovich formulation, rather than a dual formulation of the Monge problem in full generality.

The dual problem in (2.2.1) is not, in general, a value-preserving dual of the Monge problem, since $\mathcal{K}^\star$ and $\mathcal{M}^\star$ can differ. They coincide when an optimal map exists, e.g. under the hypotheses of Theorem 2.3.4 below. In general

$$\mathcal{K}^\star(\mu,\nu) \le \mathcal{M}^\star(\mu,\nu),$$

and the inequality can be strict. In Example 2.3.3 the optimal plan splits the mass of an atom — a point carrying positive mass — between two destinations, which no map can do, and the two values differ.

Example 2.3.3 (Inequality between the Kantorovich and Monge costs)**.** Consider the cost $c(x,y) = |x-y|^2$ and let

$$\mu = \frac{2}{5}\delta_{-1} + \frac{3}{5}\delta_1, \qquad \nu = \frac{3}{5}\delta_{-1} + \frac{2}{5}\delta_1.$$

Then the only admissible map sends $-1 \mapsto 1$ and $1 \mapsto -1$, so that $\mathcal{M}^\star(\mu,\nu) = 4$. However, the transport plan $\pi = \frac{2}{5}\delta_{(-1,-1)} + \frac{1}{5}\delta_{(1,-1)} + \frac{2}{5}\delta_{(1,1)}$, where $\delta_{(x,y)}$ is the Dirac mass at the point $(x,y) \in \Omega\times\Omega$, is admissible and has cost $4/5$. So the dual problem cannot serve as a strong dual of the Monge problem in general:

$$\mathcal{D}^\star(\mu,\nu) = \mathcal{K}^\star(\mu,\nu) = \frac{4}{5} < 4 = \mathcal{M}^\star(\mu,\nu).$$

Now we explain how complementary slackness leads to a deterministic optimal coupling under the structural assumption $c(x,y) = h(x-y)$, where $h \in C^1(\mathbb{R}^d)$ and strictly convex. Let $(\varphi^\star, \psi^\star)$ be

an optimal Kantorovich pair. Proposition 2.2.5 concentrates $\pi^\star$ on the contact set $\{\varphi^\star \oplus \psi^\star = c\}$. Fix (x, y) on the contact set; then dual feasibility gives, for every $\xi \in \Omega$, $c(\xi, y) - \varphi^\star(\xi) \geq \psi^\star(y)$. Hence x is a global minimizer of the mapping $\xi \mapsto c(\xi, y) - \varphi^\star(\xi)$. Assuming the differentiability of $\varphi^\star$, the first-order condition is $\nabla\varphi^\star(x) = \nabla_x c(x, y) = \nabla h(x - y)$. Since h is strictly convex, ∇h is invertible, and this relation determines y uniquely from x:

$$y = x - (\nabla h)^{-1}\big(\nabla\varphi^\star(x)\big) =: T^\star(x).$$

This computation is formal. The rigorous statement is Theorem 2.3.4, and the proof can be found in Santambrogio (2015).

Theorem 2.3.4 (Existence and uniqueness of an optimal map (Santambrogio, 2015, Theorem 1.17))**.** *Let $\Omega \subset \mathbb{R}^d$ be compact with $\mathcal{L}^d(\partial\Omega) = 0$, where $\mathcal{L}^d$ is the d-dimensional Lebesgue measure, let μ be absolutely continuous with respect to Lebesgue measure, and let $c(x, y) = h(x-y)$ with $h \in C^1(\mathbb{R}^d)$ strictly convex. Then the OT plan is unique and induced by a map, $\pi^\star = (\mathrm{id}, T^\star)_\#\mu$, where $\mathrm{id}(x) := x$ is the identity map. Moreover, if $\varphi^\star$ is optimal for the Kantorovich dual problem in* (2.2.1)*, then*

$$T^\star(x) = x - (\nabla h)^{-1}\big(\nabla\varphi^\star(x)\big) \qquad \mu\text{-a.e.} \tag{2.3.2}$$

Remark 2.3.5 (Monge as a deterministic coupling). A map $T \in \mathcal{T}(\mu, \nu)$ induces the coupling $\pi_T := (\mathrm{id}, T)_\#\mu$, i.e. the law of $(X, T(X))$ with $X \sim \mu$. Note that all the mass sits on the graph $\{(x, T(x)) : x \in \Omega\}$, a $d-$dimensional set inside the product $\Omega \times \Omega$; such a set has zero $2d$-volume, so π_T is singular with respect to Lebesgue measure. This is the special case of Remark 2.1.5 in which the conditional law $\pi_x = \delta_{T(x)}$ is a Dirac mass, so that Y is a deterministic function of X. The Monge problem (2.3.1) is therefore the restriction of the Kantorovich problem to deterministic couplings.

Corollary 2.3.6 (Brenier's theorem (Santambrogio, 2015))**.** *Specialise Theorem 2.3.4 to the quadratic cost $c(x, y) = \frac{1}{2}\|x - y\|^2$, so $h(z) = \frac{1}{2}\|z\|^2$ and $\nabla h = \mathrm{id}$. Writing $u(x) := \frac{1}{2}\|x\|^2 - \varphi^\star(x)$ for a Kantorovich potential $\varphi^\star$, the function u is convex and*

$$T^\star(x) = \nabla u(x) = x - \nabla\varphi^\star(x) \qquad \mu\text{-a.e.}, \tag{2.3.3}$$

and this $T^\star$ is the unique solution of the Monge problem (2.3.1)*. Brenier's theorem is precisely the statement that the optimal map for the quadratic cost is the gradient of a convex function (Brenier, 1991). The unscaled squared Euclidean cost $\|x - y\|^2$ has the same optimal plan and map and only doubles the transport value.*

Proposition 2.3.7 (Monge–Ampère equation)**.** *In the setting of Corollary 2.3.6, assume in addition that μ and ν are absolutely continuous with respect to Lebesgue measure, with positive continuous densities ρ_μ and ρ_ν, so that $\mathrm{d}\mu(x) = \rho_\mu(x)\,\mathrm{d}x$ and $\mathrm{d}\nu(y) = \rho_\nu(y)\,\mathrm{d}y$. Let $T^\star = \nabla u$ be the Brenier map with u convex. Then $T^\star_\#\mu = \nu$ is equivalent to the Monge–Ampère system*

$$\begin{cases} \det\big(D^2u(x)\big)\,\rho_\nu\big(\nabla u(x)\big) = \rho_\mu(x), & x \in \Omega, \\ \nabla u(\Omega) = \Omega, \end{cases}$$

the first equation holding for a.e. $x \in \Omega$.

Proof. See Santambrogio (2015, Section 1.7.6). □

Remark 2.3.8. The first equation is the change-of-variables formula for the pushforward: under the substitution $y = \nabla u(x)$ a volume element is rescaled by $\mathrm{d}y = \det\big(D^2u(x)\big)\,\mathrm{d}x$, so mass conservation $\rho_\mu(x)\,\mathrm{d}x = \rho_\nu(y)\,\mathrm{d}y$ gives exactly $\rho_\mu = \rho_\nu(\nabla u)\,\det(D^2u)$. The second equation is its global counterpart: it forces the gradient to map the source domain *onto* the target domain, so that ∇u reaches all of ν's mass rather than merely matching densities locally (Santambrogio, 2015).

2.4 Semi-dual formulation and c-transform

This section removes one of the two unknowns from the dual problem. For a potential φ, its c-transform is the largest ψ that keeps the pair (φ, ψ) admissible, so the dual becomes a maximisation over a single function. The constraint disappears in the process, since the pair (φ, φ^c) is admissible by construction. The dual-ascent solvers of Part 3 discretize this single-potential form, and the transform itself is the basic operation inside the Back-and-Forth method (Jacobs and Léger, 2020).

Definition 2.4.1 (c-transform (Santambrogio, 2015, Definition 1.10))**.** The c-transform of $\varphi : \Omega \to \mathbb{R} \cup \{+\infty\}$ is the function, denoted by φ^c, defined by

$$\varphi^c(y) := \inf_{x\in\Omega} \{\, c(x,y) - \varphi(x) \,\} \quad \forall y \in \Omega\,.$$

Definition 2.4.2 (c-concave (Santambrogio, 2015, Definition 1.10))**.** A function $\varphi : \Omega \to \mathbb{R}\cup\{+\infty\}$ is called c-concave if there exist a function $\psi : \Omega \to \mathbb{R} \cup \{+\infty\}$ such that

$$\varphi = \psi^c\,.$$

We denote by $c\text{-conc}(\Omega)$ the set of all c-concave functions on Ω.

The c-transform reduces the two-function dual problem (2.2.1) to a maximization over a single potential.

Proposition 2.4.3 (Improvement and idempotence (Santambrogio, 2015))**.** *For any φ one has $\varphi^{cc} \geq \varphi$ and $\varphi^{ccc} = \varphi^c$, where $\varphi^{cc} := (\varphi^c)^c$ with $(\varphi^c)^c(x) := \inf_{y\in\Omega}\{\, c(x,y) - \varphi^c(y) \,\}$. Consequently, φ is c-concave if and only if $\varphi = \varphi^{cc}$. Moreover, given any admissible pair (φ, ψ) for the dual problem* (2.2.1)*, the pair $(\varphi^{cc}, \varphi^c)$ is admissible and does not decrease the dual objective.*

Proof. See Santambrogio (2015, Proposition 1.34 and Theorem 1.39) or Peyré and Cuturi (2019, Section 3.2). □

Proposition 2.4.4 (Existence and semi-dual formulation (Santambrogio, 2015; Peyré and Cuturi, 2019))**.** *Let Ω be compact and c continuous. Then the dual problem* (2.2.1) *admits a maximiser of the form (φ, φ^c) with $\varphi \in c\text{-conc}(\Omega)$, and the Kantorovich value reduces to a maximisation over a single potential,*

$$\mathcal{K}^\star(\mu,\nu) = \mathcal{D}^\star(\mu,\nu) = \sup_{\varphi\in c\text{-conc}(\Omega)} \left\{ \int_\Omega \varphi\,\mathrm{d}\mu + \int_\Omega \varphi^c\,\mathrm{d}\nu \right\}. \tag{2.4.1}$$

Proof. See Santambrogio (2015, Proposition 1.11). □

Definition 2.4.5 (Kantorovich potential)**.** A maximiser φ in Proposition 2.4.4 is a *Kantorovich potential* for the transport from μ to ν (Santambrogio, 2015).

2.5 Shannon-entropy regularization of OT

The optimal plans of the previous sections are hard to compute directly. Discretized on n points, the Kantorovich problem is a linear program with n^2 unknowns. When an optimal map exists (Theorem 2.3.4), the optimal plan $\pi^\star = (\mathrm{id}, T^\star)_\# \mu$ is *singular* (Remark 2.3.5). For the quadratic cost recovering the optimal map from a transport plan π amounts to solving the Monge–Ampère equation (Proposition 2.3.7), a fully nonlinear second-order equation. Entropic regularization adds a penalty that *forbids* singular plans: it replaces $\pi^\star$ by a nearby plan π^ε that carries a density and is uniquely and stably computable, with π^ε returning to the optimal plan as $\varepsilon \to 0$.

Definition 2.5.1 (Relative entropy)**.** For two probability measures $\pi, \xi \in \mathcal{P}(\Omega \times \Omega)$, the relative entropy (Kullback–Leibler divergence) of π with respect to ξ is

$$\mathcal{H}(\pi \,\|\, \xi) := \int_{\Omega\times\Omega} \log\Big(\tfrac{\mathrm{d}\pi}{\mathrm{d}\xi}\Big)\, \mathrm{d}\pi$$

when π is absolutely continuous with respect to ξ, and $+\infty$ otherwise (Peyré and Cuturi, 2019, Rem. 4.2). Here $\mathrm{d}\pi/\mathrm{d}\xi$ is the Radon–Nikodym density of π with respect to ξ.

The regularization is built from the *Shannon entropy*. For π with density $\rho = \mathrm{d}\pi/\mathrm{d}\xi$, the relative entropy of Definition 2.5.1 is $\mathcal{H}(\pi \,\|\, \xi) = \int_{\Omega\times\Omega} \rho \log \rho \,\mathrm{d}\xi$. The integral is nonnegative by Jensen's inequality, and it vanishes if and only if $\pi = \xi$. It measures how far π is from the reference ξ, growing without bound as π concentrates onto a ξ-null set (a set of ξ-measure zero). Penalising by $\mathcal{H}$ therefore rewards spread-out plans and drives the solution away from the singular plans of Theorem 2.3.5, which carry no density. The choice is right on three counts. The function $\rho \log \rho$ is strictly convex, which gives a *unique* minimiser (Theorem 2.5.3 below). Its logarithmic structure turns the optimality conditions into the explicit Gibbs form of Theorem 2.5.3 and the Sinkhorn method of Section 3.2. And it carries the information-theoretic meaning of Theorem 2.5.5.

Definition 2.5.2 (Entropic optimal transport)**.** For a regularization strength $\varepsilon > 0$, the entropic transport objective and its optimal value are

$$\mathcal{K}_\varepsilon(\pi) := \int_{\Omega\times\Omega} c\,\mathrm{d}\pi + \varepsilon\, \mathcal{H}(\pi \,\|\, \mu \otimes \nu), \qquad \mathcal{K}^\star_\varepsilon(\mu,\nu) := \min_{\pi \in \Pi(\mu,\nu)} \mathcal{K}_\varepsilon(\pi), \tag{2.5.1}$$

the Kantorovich cost penalised by the relative entropy of the plan against the independent product $\mu \otimes \nu$ (Peyré and Cuturi, 2019, Eq. 4.9).

Proposition 2.5.3 (Uniqueness and Gibbs form)**.** *For every $\varepsilon > 0$ the objective $\mathcal{K}_\varepsilon$ is strictly convex, so the problem has a unique solution π^ε. It is the relative-entropy projection onto $\Pi(\mu,\nu)$ of the* Gibbs kernel $\mathbb{G}$*, defined through its density against $\mu \otimes \nu$,*

$$\mathrm{d}\mathbb{G}(x,y) := e^{-c(x,y)/\varepsilon}\, \mathrm{d}\mu(x)\, \mathrm{d}\nu(y), \qquad \pi^\varepsilon = \operatorname*{arg\,min}_{\pi \in \Pi(\mu,\nu)} \mathcal{H}(\pi \,\|\, \mathbb{G}),$$

and admits the form $\mathrm{d}\pi^\varepsilon(x,y) = e^{(\varphi(x)+\psi(y)-c(x,y))/\varepsilon}\, \mathrm{d}\mu(x)\, \mathrm{d}\nu(y)$ for potentials (φ,ψ) (Peyré and Cuturi, 2019, Eqs. 4.7, 4.11).

Proposition 2.5.4 (Limits in ε)**.** *As $\varepsilon \to 0$, $\mathcal{K}^\star_\varepsilon(\mu,\nu) \to \mathcal{K}^\star(\mu,\nu)$ and π^ε converges to the optimal plan of maximal entropy; as $\varepsilon \to \infty$, $\pi^\varepsilon \to \mu \otimes \nu$ (Peyré and Cuturi, 2019, Prop. 4.1).*

Remark 2.5.5 (Mutual-information reading). Read $\pi \in \Pi(\mu,\nu)$ as the joint law of (X,Y), $X \sim \mu$, $Y \sim \nu$ (Remark 2.1.5). The product $\mu \otimes \nu$ is the law of an *independent* pair, and the *(Shannon) mutual information* of the pair (X,Y) is defined as

$$I(X,Y) := \mathcal{H}(\pi \,\|\, \mu \otimes \nu),$$

so the entropy penalty is exactly the mutual information, which measures how far the coupling π is from independence — how much knowing X constrains Y. The entropic problem is then

$$\mathcal{K}^{\star}_{\varepsilon}(\mu,\nu) = \min_{\pi \in \Pi(\mu,\nu)} \mathbb{E}_{(X,Y)\sim\pi}\big[c(X,Y)\big] + \varepsilon\, I(X,Y).$$

Here $I(X,Y) \geq 0$, with $I(X,Y) = 0$ exactly when $\pi = \mu \otimes \nu$ (independence: X says nothing about Y). At the other extreme a deterministic coupling $Y = T(X)$ is singular against $\mu \otimes \nu$, so $I(X,Y) = +\infty$ (Definition 2.5.1). The penalty $\varepsilon\, I(X,Y)$ thus penalises a coupling for the dependence it carries — infinitely so for the singular optimal plans of Remark 2.3.5 — which is what forces π^{ε} to spread into a density, and, as $\varepsilon \to \infty$, all the way to the independent product (Proposition 2.5.4).

Proposition 2.5.6 (Entropic Kantorovich duality (Peyré and Cuturi, 2019))**.** *Let $\varepsilon > 0$. The entropic optimal transport problem* (2.5.1) *admits the dual formulation*

$$\begin{aligned} \mathcal{K}^{\star}_{\varepsilon}(\mu,\nu) &= \sup_{\varphi,\psi \in C(\Omega)} \mathcal{D}_{\varepsilon}(\varphi,\psi), \\ \mathcal{D}_{\varepsilon}(\varphi,\psi) &:= \int_{\Omega} \varphi \,\mathrm{d}\mu + \int_{\Omega} \psi \,\mathrm{d}\nu - \varepsilon \int_{\Omega\times\Omega} \exp\Big(\tfrac{\varphi(x)+\psi(y)-c(x,y)}{\varepsilon}\Big) \,\mathrm{d}\mu(x)\,\mathrm{d}\nu(y) \;+\; \varepsilon. \end{aligned} \tag{2.5.2}$$

The constant ε comes from the convention of Theorem 2.5.1: at any optimal pair the exponential integrates to one, so the last two terms cancel and the supremum equals $\mathcal{K}^{\star}_{\varepsilon}(\mu,\nu)$ exactly. In contrast to the unregularized dual, the hard constraint $\varphi \oplus \psi \leq c$ is replaced by the smooth exponential penalty, so the entropic dual is unconstrained, and the optimal potentials are unique up to the additive shift $(\varphi,\psi) \mapsto (\varphi + t, \psi - t)$, $t \in \mathbb{R}$.

2.6 Wasserstein distance

Fixing the cost $c(x,y) = \|x-y\|^p$ and letting the pair (μ,ν) vary turns the transport value $\mathcal{K}^{\star}(\mu,\nu)$ into a *distance* on the space of measures, making OT a tool for comparing distributions geometrically rather than pointwise.

Definition 2.6.1 (p-Wasserstein distance)**.** Let $p \geq 1$ and $c(x,y) = \|x-y\|^p$. The p-Wasserstein distance between $\mu,\nu \in \mathcal{P}(\Omega)$ is

$$W_p(\mu,\nu) := \Big(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\Omega\times\Omega} \|x-y\|^p \,\mathrm{d}\pi(x,y)\Big)^{1/p}. \tag{2.6.1}$$

For $p = 2$, W_2 is also known in statistics as the Mallows distance. On compact Ω every probability measure has finite p-th moment, so W_p is defined on all of $\mathcal{P}(\Omega)$. On unbounded domains one restricts to the finite-moment *Wasserstein space*

$$\mathcal{P}_p(\mathbb{R}^d) := \Big\{\mu \in \mathcal{P}(\mathbb{R}^d) : \int_{\mathbb{R}^d} \|x\|^p \,\mathrm{d}\mu(x) < \infty\Big\},$$

needed for $W_p < \infty$. The exponent p selects the geometry: larger p penalises long-distance displacement more heavily. The infimum is attained and W_p is finite (Santambrogio, 2015, Sec. 5.1), so $W_p(\mu, \nu)$ is well defined.

Proposition 2.6.2 (Metric axioms (Peyré and Cuturi, 2019, Prop. 2.3))**.** *On a compact Ω, W_p is a metric on $\mathcal{P}(\Omega)$: it is symmetric, nonnegative, vanishes if and only if $\mu = \nu$, and satisfies the triangle inequality*

$$W_p(\mu, \gamma) \leq W_p(\mu, \nu) + W_p(\nu, \gamma).$$

Computing W_p exactly requires solving the transport problem and is costly at scale, as quantified in Part 3. The entropic value $\mathcal{K}_\varepsilon^\star(\mu, \nu)$ of (2.5.1) is cheap to compute, but it carries a systematic offset, known in the literature as the *entropic bias*. The entropy penalty pulls the optimal coupling toward the independent product, so $\mathcal{K}_\varepsilon^\star(\mu, \mu) \neq 0$ and $\mathcal{K}_\varepsilon^\star$ is not minimised at coinciding arguments (Feydy et al., 2019). The *Sinkhorn divergence* removes this offset by subtracting the self-terms,

$$\mathrm{S}_\varepsilon(\mu, \nu) := \mathcal{K}_\varepsilon^\star(\mu, \nu) - \tfrac{1}{2}\mathcal{K}_\varepsilon^\star(\mu, \mu) - \tfrac{1}{2}\mathcal{K}_\varepsilon^\star(\nu, \nu), \tag{2.6.2}$$

so that $\mathrm{S}_\varepsilon(\mu, \mu) = 0$ by construction, which removes the bias. For the costs $c(x, y) = \|x - y\|^p$ with $0 < p < 2$, the divergence S_ε behaves like a distance: it is nonnegative and vanishes only when $\mu = \nu$ (Feydy et al., 2019).

Part 3

Computational Algorithms

This part presents the solvers that the comparison of Part 4 runs. The selection follows the formulations of Part 2, with at least one solver for the exact problem, one for the entropic dual, and one that produces a map. The simplex method solves the discrete problem exactly. The Sinkhorn algorithm and dual ascent maximize the entropic dual. The Back-and-Forth method maximizes the unregularized semi-dual and returns a map. A stochastic method closes the chapter. The material in this part is a review and follows the sources cited with each method. The contributions of the thesis — the implementations, the comparison built on them, and the barycenter method — are the subject of Part 4.

The solvers operate on discretized measures, so we fix the discrete setting first. We discretize the common domain $\Omega \subset \mathbb{R}^d$ with a finite set of source nodes $\mathcal{X} = \{x_i\}_{i=1}^n \subset \Omega$ and a finite set of target nodes $\mathcal{Y} = \{y_j\}_{j=1}^m \subset \Omega$, and approximate $\mu, \nu \in \mathcal{P}(\Omega)$ by

$$\mu = \sum_{i=1}^{n} a_i \, \delta_{x_i}, \qquad \nu = \sum_{j=1}^{m} b_j \, \delta_{y_j}, \tag{3.0.1}$$

with $a_i, b_j \geq 0$ and $\sum_{i=1}^n a_i = \sum_{j=1}^m b_j = 1$. Here n and m are finite; the discrete measures (3.0.1) are finitely supported approximations of μ, ν, and all sums below are finite. Let $\pi \in \mathbb{R}_{0,+}^{n \times m}$ denote the transport matrix representing the plan, and let $C \in \mathbb{R}^{n \times m}$ be the cost matrix associated with $c \in C_b(\Omega \times \Omega)$, given by $C = (C_{ij})_{i \in [n],\, j \in [m]}$ with $C_{ij} := c(x_i, y_j)$ for $i \in [n],\ j \in [m]$.
The admissible transport matrices are the nonnegative matrices with the prescribed marginals,

$$\Pi(a, b) := \Big\{ \pi \in \mathbb{R}_{0,+}^{n \times m} : \ \sum_{j=1}^{m} \pi_{ij} = a_i \ \ \forall i \in [n], \ \ \sum_{i=1}^{n} \pi_{ij} = b_j \ \ \forall j \in [m] \Big\}, \tag{3.0.2}$$

the matrix form of Theorem 2.1.3 for the measures (3.0.1). The discrete transport objective and its optimal value are

$$\mathcal{K}(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{m} \pi_{ij} C_{ij}, \qquad \mathcal{K}^\star = \min_{\pi \in \Pi(a,b)} \mathcal{K}(\pi), \tag{3.0.3}$$

the special case of (2.1.1) for the same measures.
Similarly, the *Shannon-entropy regularized discrete optimal transport problem* has objective and value

$$\mathcal{K}_\varepsilon(\pi) := \sum_{i=1}^{n} \sum_{j=1}^{m} \pi_{ij} C_{ij} + \varepsilon \sum_{i=1}^{n} \sum_{j=1}^{m} \pi_{ij} \big(\log \pi_{ij} - 1 \big), \qquad \mathcal{K}_\varepsilon^\star := \min_{\pi \in \Pi(a,b)} \mathcal{K}_\varepsilon(\pi). \tag{3.0.4}$$

On $\Pi(a, b)$ this penalty differs from the relative entropy $\mathcal{H}(\pi \,\|\, a \otimes b)$ of (2.5.1) by a constant fixed by the marginals, so both have the same minimiser π^ε; the shifted form $\log \pi_{ij} - 1$ is kept because it gives $\partial_{\pi_{ij}} \big[\pi_{ij} (\log \pi_{ij} - 1) \big] = \log \pi_{ij}$ and hence a cleaner dual derivation.
The problems (3.0.3) and (3.0.4) admit dual formulations. Introducing dual potentials (Lagrange multipliers) $\varphi \in \mathbb{R}^n$ and $\psi \in \mathbb{R}^m$ for the source and target constraints, the dual objective and value of (3.0.3) are

$$\mathcal{D}(\varphi, \psi) := \sum_{i=1}^{n} \varphi_i a_i + \sum_{j=1}^{m} \psi_j b_j, \qquad \mathcal{D}^\star := \max_{\varphi \oplus \psi \leq C} \mathcal{D}(\varphi, \psi),$$

where $\varphi \oplus \psi \leq C$ abbreviates $\varphi_i + \psi_j \leq C_{ij}$ for all $i \in [n]$, $j \in [m]$.

For the regularized problem (3.0.4) the dual objective and value are

$$\mathcal{D}_\varepsilon(\varphi,\psi) := \sum_{i=1}^{n}\varphi_i a_i + \sum_{j=1}^{m}\psi_j b_j - \varepsilon\sum_{i=1}^{n}\sum_{j=1}^{m}\exp\Big(\tfrac{\varphi_i+\psi_j-C_{ij}}{\varepsilon}\Big), \qquad \mathcal{D}_\varepsilon^\star := \max_{\varphi\in\mathbb{R}^n,\,\psi\in\mathbb{R}^m}\mathcal{D}_\varepsilon(\varphi,\psi). \tag{3.0.5}$$

Note that the hard constraint $\varphi\oplus\psi\le C$ of the unregularized dual is replaced here by the smooth exponential penalty, so the entropic dual is unconstrained.

3.1 Linear programming

The primal problem (3.0.3) is an instance of a linear program. To recast it in canonical LP form we vectorize the matrices, using the same (row-major) ordering for both so that

$$\gamma = \mathrm{vec}(\pi)\in\mathbb{R}^{nm}_{0,+}, \qquad c = \mathrm{vec}(C)\in\mathbb{R}^{nm}_{0,+}, \qquad c^\top\gamma = \sum_{i,j}C_{ij}\pi_{ij} = \langle C,\pi\rangle,$$

i.e., the LP objective is exactly the transport cost. The discrete OT problem then reads, in standard form,

$$\min_{\gamma\in\mathbb{R}^{nm}_{0,+}} c^\top\gamma \qquad \text{s.t.} \qquad A\gamma = p := \begin{pmatrix} a\\ b\end{pmatrix},$$

where $A = \big[\, I_n\otimes\mathbf{1}_m^\top;\ \mathbf{1}_n^\top\otimes I_m\,\big]$ is the $(n+m)\times nm$ constraint matrix whose top block sums each row of π and whose bottom block sums each column. The generic two-phase simplex method of Appendix A.1 is stated for a maximisation; the minimisation above is the same problem under $c\mapsto -c$.

The $n+m$ marginal equations are not independent: summing the n row equations and summing the m column equations both give the total-mass equation $\sum_{i,j}\pi_{ij}=1$, so one equation is redundant and, for positive marginals, $\mathrm{rank}(A) = n+m-1 =: r$. We drop one redundant equation, leaving a system of full row rank r. By the rank–nullity theorem, since $A\gamma=p$ is consistent there is an index set $B=\{j_1,\dots,j_r\}\subset[nm]$ whose columns $A_{[j_1]},\dots,A_{[j_r]}$ are linearly independent, so the corresponding basic solution has at most r nonzero entries. This partitions the variable indices into the *basic* set B and the *non-basic* set $N := [nm]\setminus B$:

$$[nm] = B\cup N, \quad B\cap N = \varnothing, \qquad \gamma_{[B]}\in\mathbb{R}^r_{0,+}, \qquad \gamma_{[N]}\in\mathbb{R}^{nm-r}_{0,+}.$$

Geometrically, the feasible set $\{\gamma\in\mathbb{R}^{nm}_{0,+} : A\gamma = p\}$ is *bounded*: every entry obeys $0\le\pi_{ij}\le\min(a_i,b_j)$, so no coordinate escapes to infinity. It is a polytope of dimension $nm-r=(n-1)(m-1)$, with infinitely many points but finitely many extreme points, and these extreme points form the search space of the simplex method (Peyré and Cuturi, 2019, Chapter 3). An extreme point is a *basic feasible solution* (BFS): it has at most $r=n+m-1$ positive entries ($nm-r$ zeros), exactly r in the nondegenerate case.

The simplex method walks from one BFS to an adjacent one, each step strictly decreasing $c^\top\gamma$, and stops once no adjacent BFS lowers the cost. The plan $\pi^\star$ it returns is therefore both *exact* (unregularized) and *sparse*, with at most $n+m-1$ nonzero entries. The generic two-phase construction is recalled in Appendix A.1.

3.2 Sinkhorn algorithm

The Sinkhorn algorithm provides an efficient and stable way to solve the Shannon-entropy regularized OT problem (3.0.4). The procedure was introduced by Cuturi (2013) in the context of OT, while the convergence proof is attributed to Sinkhorn and Knopp (1967).

The Sinkhorn algorithm exploits the shape of the solution π^ε of the Shannon-entropy regularized OT problem (3.0.4),

$$\pi^\varepsilon = (\pi^\varepsilon_{ij})_{i\in[n],j\in[m]}, \text{ with } \quad \pi^\varepsilon_{ij} = \exp\left\{\frac{\varphi_i + \psi_j - C_{ij}}{\varepsilon}\right\}, \forall\,(i,j) \in [n]\times[m], \varepsilon > 0,$$

where φ_i and ψ_j are the i-th and j-th components of the dual potentials φ and ψ of (3.0.5). The potentials depend on ε, and we write φ, ψ instead of $\varphi^\varepsilon, \psi^\varepsilon$ to keep the notation light. The solution π^ε must also satisfy the marginal constraints of (3.0.2), which forces

$$a_i = \sum_{j=1}^{m} \exp\left\{\frac{\varphi_i + \psi_j - C_{ij}}{\varepsilon}\right\} \quad \text{and} \quad b_j = \sum_{i=1}^{n} \exp\left\{\frac{\varphi_i + \psi_j - C_{ij}}{\varepsilon}\right\}.$$

Solving the first equation for φ_i and the second for ψ_j gives

$$\varphi_i = \varepsilon \ln(a_i) - \varepsilon \ln\left(\sum_{j=1}^{m} \exp\left\{\frac{\psi_j - C_{ij}}{\varepsilon}\right\}\right)$$

and

$$\psi_j = \varepsilon \ln(b_j) - \varepsilon \ln\left(\sum_{i=1}^{n} \exp\left\{\frac{\varphi_i - C_{ij}}{\varepsilon}\right\}\right).$$

The Sinkhorn algorithm iterates these two equations. Initialize $\psi^{(0)} = 0$ and define the iterative sequences $(\varphi^{(k)})_{k\geq 0}$ and $(\psi^{(k)})_{k\geq 0}$ as follows

$$\varphi_i^{(k+1)} = \varepsilon \ln(a_i) - \varepsilon \ln\left(\sum_{j=1}^{m} \exp\left\{\frac{\psi_j^{(k)} - C_{ij}}{\varepsilon}\right\}\right), \tag{3.2.1}$$

$$\psi_j^{(k+1)} = \varepsilon \ln(b_j) - \varepsilon \ln\left(\sum_{i=1}^{n} \exp\left\{\frac{\varphi_i^{(k+1)} - C_{ij}}{\varepsilon}\right\}\right), \tag{3.2.2}$$

where the superscript (k) denotes the values of the dual variables at the k-th iteration. The iteration stops once the marginal error of the induced plan falls below a tolerance τ, as summarized in Algorithm 1 and discussed in Remark 3.2.1.

Remark 3.2.1. For monitoring the convergence, the marginal errors $\|\pi^\varepsilon \mathbf{1}_m - a\|_2$ and $\|(\pi^\varepsilon)^\top \mathbf{1}_n - b\|_2$ are a useful stopping criterion (Cuturi, 2013). Algorithm 1 stops when the larger of the two falls below the tolerance τ.

Sinkhorn admits *local convergence* with rate $\kappa \sim \varepsilon$ near the optimum (Peyré and Cuturi, 2019,

Algorithm 1 Sinkhorn Algorithm

Input: Cost matrix $C \in \mathbb{R}_+^{n\times m}$, probability vectors $a \in \mathbb{R}_+^n$ and $b \in \mathbb{R}_+^m$, regularization parameter $\varepsilon > 0$, marginal tolerance τ

Initialize: $\varphi^{(0)} \leftarrow (0,\dots,0)_n$, $\psi^{(0)} \leftarrow (0,\dots,0)_m$

repeat

$\varphi^{(k+1)} \leftarrow \varepsilon\left(\ln(a) - \operatorname{logsumexp}\left\{\frac{\mathbf{1}_n(\psi^{(k)})^\top - C}{\varepsilon}\right\}\right)$

$\psi^{(k+1)} \leftarrow \varepsilon\left(\ln(b) - \operatorname{logsumexp}\left\{\frac{\varphi^{(k+1)}\mathbf{1}_m^\top - C}{\varepsilon}\right\}\right)$

$\pi^\varepsilon \leftarrow \exp\left(\frac{\varphi^{(k+1)}\mathbf{1}_m^\top + \mathbf{1}_n(\psi^{(k+1)})^\top - C}{\varepsilon}\right)$

$k \leftarrow k+1$

until $\max\left\{\ \|\pi^\varepsilon \mathbf{1}_m - a\|_2\,,\ \|(\pi^\varepsilon)^\top \mathbf{1}_n - b\|_2\ \right\} \le \tau$

return $\varphi, \psi, \pi^\varepsilon$

Remark 4.15). Reaching a tolerance δ requires $O(\log(1/\delta)/\varepsilon)$ iterations:

$$\|\varphi^{(k)} - \varphi^*\| \le (1-\kappa)^k \|\varphi^{(0)} - \varphi^*\| \Longrightarrow k = O\left(\frac{\log(1/\delta)}{-\log(1-\kappa)}\right) \underset{\kappa \ll 1}{=} O\left(\frac{\log(1/\delta)}{\kappa}\right) \underset{\kappa \sim \varepsilon}{=} O\left(\frac{\log(1/\delta)}{\varepsilon}\right).$$

3.3 Sinkhorn algorithm as an alternate maximization of the dual problem

The entropic dual (3.0.5) is smooth and unconstrained, so it can be maximized by plain gradient ascent, and the Sinkhorn algorithm turns out to be the implicit version of that ascent. The derivative of the dual objective (3.0.5) with respect to the dual variables φ_i and ψ_j for all $i \in [n]$ and $j \in [m]$ is given by

$$\mathcal{G}_{\varepsilon,\varphi,i} := \frac{\partial \mathcal{D}^\varepsilon}{\partial \varphi_i}(\varphi,\psi) = a_i - \sum_{j=1}^m \exp\left\{\frac{\varphi_i + \psi_j - C_{ij}}{\varepsilon}\right\} \quad \in \mathbb{R}, \tag{3.3.1}$$

$$\mathcal{G}_{\varepsilon,\psi,j} := \frac{\partial \mathcal{D}^\varepsilon}{\partial \psi_j}(\varphi,\psi) = b_j - \sum_{i=1}^n \exp\left\{\frac{\varphi_i + \psi_j - C_{ij}}{\varepsilon}\right\} \quad \in \mathbb{R}. \tag{3.3.2}$$

Let $\varphi_i^{(0)} = 0$, $\forall i \in [n]$ and $\psi_j^{(0)} = 0$, $\forall j \in [m]$, then the gradient ascent algorithm is defined as follows:

$$\varphi_i^{(k+1)} = \varphi_i^{(k)} + \eta\, \mathcal{G}_{\varepsilon,\varphi,i}^{(k)}, \quad \forall i \in [n], \tag{3.3.3}$$

$$\psi_j^{(k+1)} = \psi_j^{(k)} + \eta\, \mathcal{G}_{\varepsilon,\psi,j}^{(k)}, \quad \forall j \in [m], \tag{3.3.4}$$

where $\eta > 0$ is the step size (learning rate) and the superscript (k) on $\mathcal{G}_{\varepsilon,\varphi,i}$ and $\mathcal{G}_{\varepsilon,\psi,j}$ means that the gradients of (3.3.1) and (3.3.2) are evaluated at the current iterates $\varphi^{(k)}$ and $\psi^{(k)}$. Maximizing the smoothed entropic dual with first-order and quasi-Newton methods was proposed by Cuturi and Peyré (2016).

Notice that the system of equations defined in (3.3.1) and (3.3.2) is equivalent to the Sinkhorn algorithm defined in (3.2.1). Setting the gradient (3.3.1) to zero gives

$$a_i = \sum_{j=1}^m \exp\left\{\frac{\varphi_i + \psi_j - C_{ij}}{\varepsilon}\right\}, \qquad \text{that is,} \qquad \exp\left\{\frac{\varphi_i}{\varepsilon}\right\} = \frac{a_i}{\sum_{j=1}^m \exp\left\{\frac{\psi_j - C_{ij}}{\varepsilon}\right\}},$$

and therefore

$$\varphi_i = \varepsilon \ln a_i - \varepsilon \ln \sum_{j=1}^{m} \exp\left\{\frac{\psi_j - C_{ij}}{\varepsilon}\right\},$$

which is the update operation for φ_i in the Sinkhorn algorithm (3.2.1). Hence, the Sinkhorn algorithm can be interpreted as the fixed-point or implicit version of the gradient ascent, where instead of using the finite step η one sets the gradient to zero at each iteration.

Remark 3.3.1. Notice that the Sinkhorn algorithm could be defined more abstractly as

$$\varphi^{(k+1)} \in \operatorname*{argmax}_{\varphi}\{\mathcal{D}^{\varepsilon}(\varphi, \psi^{(k)})\}, \qquad \psi^{(k+1)} \in \operatorname*{argmax}_{\psi}\{\mathcal{D}^{\varepsilon}(\varphi^{(k+1)}, \psi)\}, \tag{3.3.5}$$

where each block maximiser is given in closed form by the entropic (soft) c-transform. Solving $\partial_{\varphi_i}\mathcal{D}^{\varepsilon} = 0$ yields

$$\varphi_i^{(k+1)} = \left(\psi^{(k)}\right)_i^{(c,\varepsilon)} := \varepsilon \ln a_i - \varepsilon \ln \sum_{j=1}^{m} \exp\Big\{\frac{\psi_j^{(k)} - C_{ij}}{\varepsilon}\Big\} = \varepsilon \ln a_i + \operatorname{smin}_{\varepsilon}^{j}\left\{C_{ij} - \psi_j^{(k)}\right\},$$

and analogously $\psi_j^{(k+1)} = \varepsilon \ln b_j + \operatorname{smin}_{\varepsilon}^{i}\{C_{ij} - \varphi_i^{(k+1)}\}$, where $\operatorname{smin}_{\varepsilon}^{j}\{z_j\} := -\varepsilon \ln \sum_j \exp\{-z_j/\varepsilon\} \to \min_j z_j$ as $\varepsilon \to 0^+$. Thus (3.3.5) is the alternating application of soft (entropic) c-transforms, that is, the ε-smoothed analogue of the Kantorovich c-transform. This is the viewpoint adopted by Di Marino and Gerolin (2020).

Remark 3.3.2 (Adaptive first-order methods). The simple gradient ascent described above can be improved using adaptive first-order methods such as Adam (Kingma and Ba, 2015), which maintains exponential moving averages of the gradients and their squares used to estimate the momentum and for coordinatewise normalization respectively. We present Adam's update rule for one variable φ, the other variable ψ is updated analogously (Kingma and Ba, 2015). Let $m^{(k)}$ be the exponentially weighted average of the gradient (the first moment) for potential $\varphi^{(k)}$ and $v^{(k)}$ the exponentially weighted average of its square (the second moment), with the bias corrections $\hat{m}^{(k)}$ and $\hat{v}^{(k)}$, where $m^{(0)} = 0, v^{(0)} = 0$:

$$m_i^{(k)} = \beta_1 m_i^{(k-1)} + (1-\beta_1)\, \mathcal{G}_{\varepsilon,\varphi,i}^{(k)}, \quad v_i^{(k)} = \beta_2 v_i^{(k-1)} + (1-\beta_2)\, \left(\mathcal{G}_{\varepsilon,\varphi,i}^{(k)}\right)^2,$$

$$\hat{m}_i^{(k)} = \frac{m_i^{(k)}}{1-\beta_1^k}, \quad \hat{v}_i^{(k)} = \frac{v_i^{(k)}}{1-\beta_2^k},$$

where $\beta_1, \beta_2 \in (0,1)$ are the decay coefficients. Then the update step is formulated as

$$\varphi_i^{(k+1)} = \varphi_i^{(k)} + \eta \frac{\hat{m}_i^{(k)}}{\sqrt{\hat{v}_i^{(k)}} + \kappa}, \tag{3.3.6}$$

where $\kappa > 0$ is the small positive scalar constant for numerical stability and $\eta > 0$ is the step size.

3.4 Back-and-Forth method

The main idea of the *Back-and-Forth* method introduced by Jacobs and Léger (2020) is to compute a solution of the dual problem (2.2.1) via the two alternate maximizations

$$\varphi \in \operatorname*{argmax}_{\varphi} \mathcal{D}(\varphi, \psi)\,, \qquad \psi \in \operatorname*{argmax}_{\psi} \mathcal{D}(\varphi, \psi)\,. \tag{3.4.1}$$

In this sense, the Back-and-Forth method can be interpreted as the analogue of the Sinkhorn algorithm in the limit when $\varepsilon \to 0^+$ (see Remark 3.3.1). The main limitation of this approach lies in the property of the c-transform procedure. If φ is a c-concave function, then one has $(\varphi^c)^c = \varphi$, which prevents the procedure (3.4.1) from increasing the value of the dual $\mathcal{D}(\varphi, \psi)$ after two iterations.

Thus one may introduce the dual functionals $J : C(\Omega) \to \mathbb{R}$ and $I : C(\Omega) \to \mathbb{R}$ depending on either of the Kantorovich potentials φ or ψ:

$$J(\varphi) = \int_\Omega \varphi \,\mathrm{d}\mu + \int_\Omega \varphi^c \,\mathrm{d}\nu, \qquad I(\psi) = \int_\Omega \psi \,\mathrm{d}\nu + \int_\Omega \psi^c \,\mathrm{d}\mu, \tag{3.4.2}$$

where $(\cdot)^c$ is the c-transform operation (see Definition 2.4.1), so that J is exactly the semi-dual objective of (2.4.1), which allows to rephrase the equations (3.4.1) as

$$\varphi \in \operatorname*{argmax}_{\varphi} J(\varphi) \qquad \psi \in \operatorname*{argmax}_{\psi} I(\psi)\,.$$

As shown in Theorem 1 of Jacobs and Léger (2020), $J(\varphi)$ is concave in φ and is maximized by a c-concave function φ^c; the same holds for $I(\psi)$, maximized by a c-concave ψ^c. One may therefore perturb the potentials in the direction that increases the dual, i.e., along its gradient:

$$\varphi \longleftarrow \varphi + \eta\, \text{“}\nabla\text{”} J(\varphi)\,.$$

3.4.1 Sobolev gradient step

As observed by Jacobs and Léger (2020), the L^2-gradient of J (respectively of I) is the raw marginal residual, whose high-frequency content makes plain L^2 ascent unstable; measuring the gradient in the $\dot{H}^1$ metric instead applies $(-\Delta)^{-1}$ to that residual, damping high frequencies and yielding a smoother, more stable ascent direction.

Definition 3.4.1 ($\dot{H}^1$-space)**.** Let $\Omega \subset \mathbb{R}^d$ be a convex and compact set. The homogeneous Sobolev space of zero-mean functions with square-integrable gradient on Ω is defined as

$$\dot{H}^1(\Omega) = \left\{ \varphi : \Omega \to \mathbb{R} : \int_\Omega \varphi(x)\,\mathrm{d}x = 0;\;\; \nabla\varphi \in L^2(\Omega) \right\}, \qquad \langle \varphi_1, \varphi_2 \rangle_{\dot{H}^1} = \int_\Omega \nabla\varphi_1(x) \cdot \nabla\varphi_2(x)\,\mathrm{d}x\,.$$

The space $\dot{H}^1(\Omega)$ is a Hilbert space; we denote its dual by $\dot{H}^{-1}(\Omega) := \big(\dot{H}^1(\Omega)\big)^*$, the space of continuous linear functionals on $\dot{H}^1(\Omega)$. By the Riesz representation theorem (Brezis, 2011, Theorem 5.5), every $\rho \in \dot{H}^{-1}(\Omega)$ has a unique representative $\varphi_\rho \in \dot{H}^1(\Omega)$ with

$$\rho(h) = \langle \varphi_\rho, h \rangle_{\dot{H}^1} = \int_\Omega \nabla\varphi_\rho(x) \cdot \nabla h(x)\,\mathrm{d}x \qquad \forall h \in \dot{H}^1(\Omega).$$

In weak form, φ_ρ is the unique zero-mean solution of the Neumann–Poisson problem

$$-\Delta\varphi_\rho = \rho \ \text{ in } \Omega, \qquad \partial_n\varphi_\rho = 0 \ \text{ on } \partial\Omega, \qquad \int_\Omega \varphi_\rho \,\mathrm{d}x = 0.$$

The first two conditions determine φ_ρ up to an additive constant. The zero-mean condition fixes that constant.

Definition 3.4.2 (Gâteaux and Fréchet differentiability)**.** Let $F\colon H^1(\Omega) \to \mathbb{R}$.

1. F is *Gâteaux differentiable* at $\varphi \in H^1(\Omega)$ if the directional derivative

$$\delta F(\varphi)[h] := \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} F(\varphi + \varepsilon h) = \lim_{\varepsilon\to 0} \frac{F(\varphi+\varepsilon h) - F(\varphi)}{\varepsilon}$$

   exists for every $h \in H^1(\Omega)$ and $h \mapsto \delta F(\varphi)[h]$ is a bounded linear functional on $H^1(\Omega)$. We call $\delta F(\varphi)[h]$ the *first variation* of F at φ in the direction h.

2. F is *Fréchet differentiable* at φ if there is a bounded linear functional $DF(\varphi)$ on $H^1(\Omega)$ with

$$F(\varphi + h) = F(\varphi) + DF(\varphi)[h] + o\big(\|h\|_{H^1}\big), \qquad \|h\|_{H^1} \to 0.$$

Fréchet differentiability implies Gâteaux differentiability, and then $DF(\varphi)[h] = \delta F(\varphi)[h]$ (Luenberger, 1969, Section 7.2, Proposition 2).

Definition 3.4.3. For a Fréchet-differentiable functional $F : \dot H^1(\Omega) \to \mathbb{R}$, its $\dot H^1$-gradient is the unique element $\nabla_{\dot H^1} F(\varphi) \in \dot H^1(\Omega)$ such that

$$\delta F(\varphi)[h] = \langle \nabla_{\dot H^1} F(\varphi),\, h\rangle_{\dot H^1} \qquad \forall h \in \dot H^1(\Omega).$$

Proposition 3.4.4 (First variation of the dual functionals Jacobs and Léger (2020, Lemma 3))**.** *Let $\Omega \subset \mathbb{R}^d$ be convex and compact and let $\mu, \nu \in \mathcal{P}(\Omega)$. Then the dual functionals $J, I : \dot H^1(\Omega) \to \mathbb{R}$ of* (3.4.2) *are Gâteaux differentiable, and for every direction $h \in \dot H^1(\Omega)$*

$$\delta J(\varphi)[h] = \int_\Omega h \,\mathrm{d}\big(\mu - (S_\varphi)_\#\nu\big), \qquad \delta I(\psi)[h] = \int_\Omega h \,\mathrm{d}\big(\nu - (T_\psi)_\#\mu\big),$$

where S_φ and T_ψ are the transport maps associated with φ and ψ, respectively. Equivalently, as elements of the dual space $\dot H^{-1}(\Omega)$,

$$\delta J(\varphi) = \mu - (S_\varphi)_\#\nu \in \dot H^{-1}(\Omega), \qquad \delta I(\psi) = \nu - (T_\psi)_\#\mu \in \dot H^{-1}(\Omega),$$

and the corresponding $\dot H^1$-gradients (their Riesz representatives) solve the Neumann–Poisson problems

$$\nabla_{\dot H^1} J(\varphi) = (-\Delta)^{-1}\big(\mu - (S_\varphi)_\#\nu\big), \qquad \nabla_{\dot H^1} I(\psi) = (-\Delta)^{-1}\big(\nu - (T_\psi)_\#\mu\big).$$

In particular, the first variations are independent of the costs. The costs enter only through the maps S_φ, T_ψ, made explicit in Section 3.4.3.

3.4.2 Computational algorithm

Consider the sequences of potentials $\{\varphi^{(k)}\}_{k\geq 0}$ and $\{\psi^{(k)}\}_{k\geq 0}$. Then the Back-and-Forth scheme is defined as

$$\varphi^{(k+1/2)} \longleftarrow \varphi^{(k)} + \sigma_k \nabla_{\dot{H}^1} J(\varphi^{(k)})\,; \tag{3.4.3a}$$

$$\psi^{(k+1/2)} \longleftarrow (\varphi^{(k+1/2)})^c\,; \tag{3.4.3b}$$

$$\psi^{(k+1)} \longleftarrow \psi^{(k+1/2)} + \sigma_k \nabla_{\dot{H}^1} I(\psi^{(k+1/2)})\,; \tag{3.4.3c}$$

$$\varphi^{(k+1)} \longleftarrow (\psi^{(k+1)})^c\,. \tag{3.4.3d}$$

Remark 3.4.5 (Well-definedness of the $\dot{H}^1$-gradients). A gradient-ascent step taken in φ alone (or in ψ alone) does not, in general, keep the iterate c-concave. If c-concavity is lost, the induced maps S_φ, T_ψ need not be well-defined, and consequently the gradients $\nabla_{\dot{H}^1} J(\varphi)$ and $\nabla_{\dot{H}^1} I(\psi)$ may fail to be well-defined as well (Jacobs and Léger, 2020). The Back-and-Forth scheme avoids this: at every step where a gradient is taken, the active potential has just been produced as the c-transform of the previous iterate (steps (3.4.3a), (3.4.3c)). Being a c-transform, it is c-concave, so the associated map and Sobolev gradient are well-defined.

Theorem 3.4.6 (Conditional monotonicity of Back-and-Forth, Jacobs and Léger (2020, Propositions 1 and 2))**.** *Let $c(x,y) = \frac{1}{2}\|x-y\|^2$ and suppose every c-concave iterate produced by the scheme* (3.4.3a)–(3.4.3d) *remains in the uniformly Hessian-bounded class*

$$S_\lambda := \Big\{\varphi \in \dot{H}^1(\Omega) : (1-\lambda^{-1})I \preceq D^2\varphi(x) \preceq (1-\lambda)I\Big\}$$

for some $\lambda \in (0,1)$. Then each $\dot{H}^1$-gradient-ascent step is an ascent step for the dual functional; together with the c-transform steps, which can only increase the dual value, the scheme is monotone:

$$\mathcal{D}\big(\varphi^{(k+1)}, \psi^{(k+1)}\big) \;\geq\; \mathcal{D}\big(\varphi^{(k+1/2)}, \psi^{(k+1/2)}\big) \;\geq\; \mathcal{D}\big(\varphi^{(k)}, \psi^{(k)}\big).$$

Remark 3.4.7 (Convergence). The monotonicity of Theorem 3.4.6 is conditional on the curvature control $\varphi^{(k)}, \psi^{(k)} \in S_\lambda$. A convergence proof of the Back-and-Forth scheme *without* such a Hessian-bound assumption on the iterates is, to our knowledge, not currently available; we refer the reader to Jacobs and Léger (2020) for a detailed discussion.

3.4.3 Practical implementation

Although the scheme is conceptually simple, each iteration relies on three numerical primitives: (i) a c-transform, (ii) the induced pushforward (Monge) map, and (iii) the $\dot{H}^1$-gradient, i.e., a Poisson solve. Algorithm 2 states the scheme in terms of these primitives, which we describe in turn. Lines 3 and 7 of Algorithm 2 build the pushforward maps, lines 4 and 8 assemble the Sobolev gradients through a Poisson solve, and the remaining lines carry out the ascent and c-transform steps of (3.4.3). Throughout we restrict to separable costs

$$c(x,y) = \sum_{k=1}^{d} h_k(y_k - x_k), \qquad h_k : \mathbb{R} \to \mathbb{R} \text{ strictly convex and even}, \tag{3.4.4}$$

which includes the quadratic cost $c(x,y) = \frac{1}{2}\|x-y\|^2$ and makes all three primitives cheap.

Algorithm 2 Back-and-Forth Algorithm

Input: densities μ, ν; step sizes $\{\sigma_k\}_{k=0,\dots,K-1}$; number of iterations K

Initialize: $\varphi^{(0)} \leftarrow 0,\ \psi^{(0)} \leftarrow 0$
for $k = 0, 1, \dots, K-1$ **do**
  $S_\varphi \leftarrow \mathrm{id} - (\nabla h)^{-1} \circ \nabla[(\varphi^{(k)})^c]$ ▷ pushes ν
  $\nabla_{\dot{H}^1} J(\varphi^{(k)}) \leftarrow (-\Delta)^{-1}\big(\mu - (S_\varphi)_\# \nu\big)$ ▷ ascent in $J(\varphi)$
  $\varphi^{(k+1/2)} \leftarrow \varphi^{(k)} + \sigma_k \nabla_{\dot{H}^1} J(\varphi^{(k)})$
  $\psi^{(k+1/2)} \leftarrow \big(\varphi^{(k+1/2)}\big)^c$
  $T_\psi \leftarrow \mathrm{id} - (\nabla h)^{-1} \circ \nabla[(\psi^{(k+1/2)})^c]$ ▷ pushes μ
  $\nabla_{\dot{H}^1} I(\psi^{(k+1/2)}) \leftarrow (-\Delta)^{-1}\big(\nu - (T_\psi)_\# \mu\big)$ ▷ ascent in $I(\psi)$
  $\psi^{(k+1)} \leftarrow \psi^{(k+1/2)} + \sigma_k \nabla_{\dot{H}^1} I(\psi^{(k+1/2)})$
  $\varphi^{(k+1)} \leftarrow \big(\psi^{(k+1)}\big)^c$
end for
return $\varphi^{(K)}, \psi^{(K)}$

The pushforward map

For a c-concave potential, the envelope condition for the c-transform yields the induced maps

$$S_\varphi(y) = y - (\nabla h)^{-1}\big(\nabla \varphi^c(y)\big), \qquad T_\psi(x) = x - (\nabla h)^{-1}\big(\nabla \psi^c(x)\big), \tag{3.4.5}$$

pushing $\nu \to \mu$ and $\mu \to \nu$ respectively. Under (3.4.4) the inverse $(\nabla h)^{-1}$ acts coordinatewise, $(\nabla h)^{-1}(z)_k = (h_k')^{-1}(z_k)$, so the maps are available in closed form whenever the $(h_k')^{-1}$ are; the method is most efficient on such costs. For the quadratic cost $c(x,y) = \frac{1}{2}\|x-y\|^2$ one has $\nabla h = \mathrm{id}$, and (3.4.5) collapses to

$$S_\varphi(y) = y - \nabla \varphi^c(y), \qquad T_\psi(x) = x - \nabla \psi^c(x).$$

Computing the c-transform

The separable structure of (3.4.4) reduces the d-dimensional c-transform

$$\varphi^c(y) = \inf_{x \in \Omega} \Big\{ \sum_{k=1}^{d} h_k(y_k - x_k) - \varphi(x) \Big\}$$

to d one-dimensional transforms applied along the axes in turn. Denote the one-dimensional c-transform along axis k (all other coordinates held fixed) by

$$\mathcal{C}_k[g](\dots, y_k, \dots) := \inf_{x_k} \big\{ h_k(y_k - x_k) - g(\dots, x_k, \dots) \big\}.$$

Peeling off one axis at a time gives the recursion

$$u_d := \varphi, \qquad u_{k-1} := -\,\mathcal{C}_k[u_k] \ \ (k = d, \dots, 2), \qquad \varphi^c = \mathcal{C}_1[u_1],$$

i.e., d successive one-dimensional sweeps, the partial result being negated between consecutive axes. Each sweep is a discrete Legendre-type transform and can be computed in $O(n)$ per line by a lower-envelope algorithm (Jacobs and Léger, 2020; Lucet, 1997), so a full c-transform on an n^d grid costs $O(d\, n^d)$. This axis-by-axis evaluation mirrors the separable structure exploited by multidimensional fast transforms.

The $\dot{H}^1$-gradient and the Poisson solve

By Proposition 3.4.4, the ascent direction is the Riesz representative $(-\Delta)^{-1}$ of the marginal residual: $\nabla_{\dot{H}^1} J(\varphi)$ is the zero-mean solution g of the Neumann–Poisson problem

$$-\Delta g = \rho \ \text{ in } \Omega, \qquad \partial_n g = 0 \ \text{ on } \partial\Omega, \qquad \int_\Omega g \,\mathrm{d}x = 0, \qquad \rho := \mu - (S_\varphi)_\# \nu, \tag{3.4.6}$$

and analogously for I with $\rho = \nu - (T_\psi)_\# \mu$. We solve (3.4.6) spectrally.

The point is to diagonalize $-\Delta$: in a suitable basis it acts by multiplication, and inversion becomes a division. The Neumann (no-flux) condition fixes the basis to be *cosines*. On the box $\Omega = [0, L]^d$ the Neumann Laplacian is diagonalized by

$$e_{\mathbf{k}}(x) = \prod_{j=1}^{d} \cos\Big(\frac{\pi k_j x_j}{L}\Big), \qquad -\Delta e_{\mathbf{k}} = \lambda_{\mathbf{k}}\, e_{\mathbf{k}}, \quad \lambda_{\mathbf{k}} = \sum_{j=1}^{d} \Big(\frac{\pi k_j}{L}\Big)^2, \qquad \mathbf{k} \in \mathbb{N}_0^d,$$

each $e_{\mathbf{k}}$ satisfying $\partial_n e_{\mathbf{k}} = 0$ on $\partial\Omega$. Expanding $g = \sum_{\mathbf{k}} \hat{g}_{\mathbf{k}} e_{\mathbf{k}}$ and $\rho = \sum_{\mathbf{k}} \hat{\rho}_{\mathbf{k}} e_{\mathbf{k}}$, (3.4.6) becomes diagonal,

$$\lambda_{\mathbf{k}}\, \hat{g}_{\mathbf{k}} = \hat{\rho}_{\mathbf{k}} \qquad \Longrightarrow \qquad \hat{g}_{\mathbf{k}} = \frac{\hat{\rho}_{\mathbf{k}}}{\lambda_{\mathbf{k}}} \ \ (\mathbf{k} \neq \mathbf{0}), \qquad \hat{g}_{\mathbf{0}} = 0. \tag{3.4.7}$$

The mode $\mathbf{k} = \mathbf{0}$ has $\lambda_{\mathbf{0}} = 0$, so the equation reads $\hat{\rho}_{\mathbf{0}} = 0$, i.e., $\int_\Omega \rho \,\mathrm{d}x = 0$, which holds because ρ is a difference of probability measures. A Neumann solution is unique only up to a constant, and setting $\hat{g}_{\mathbf{0}} = 0$ selects the zero-mean $\dot{H}^1$-representative.

In practice the cosine coefficients are computed by the discrete cosine transform (DCT), the FFT-family transform adapted to Neumann boundaries, so the gradient is obtained as

$$g = \mathrm{DCT}^{-1}\Big(\lambda^{-1}\, \mathrm{DCT}(\rho)\Big), \qquad \text{at cost } \ O(N \log N), \quad N = n^d.$$

Here $\mathrm{DCT}(\rho)$ collects the coefficients $\hat{\rho}_{\mathbf{k}}$, and λ^{-1} acts mode by mode, dividing the $\mathbf{k}$-th coefficient by $\lambda_{\mathbf{k}}$ as in (3.4.7). On the n-point grid with spacing $h = L/n$, inverting the standard three-point Laplacian *exactly* amounts to replacing $\lambda_{\mathbf{k}}$ by the discrete eigenvalue $\lambda^h_{\mathbf{k}} = \sum_j \frac{4}{h^2} \sin^2\big(\frac{\pi k_j}{2n}\big)$, which agrees with $\lambda_{\mathbf{k}}$ at low frequencies.

Step size

The step sizes σ_k are chosen by an Armijo–Goldstein line search (Armijo, 1966; Goldstein, 1965), adapted to the ascent of J (and analogously I). With ascent direction $p = \nabla_{\dot{H}^1} J(\varphi^{(k)})$, the first-order predicted gain along p is

$$\delta J(\varphi^{(k)})[p] = \big\langle \nabla_{\dot{H}^1} J(\varphi^{(k)}),\, p \big\rangle_{\dot{H}^1} = \big\| \nabla_{\dot{H}^1} J(\varphi^{(k)}) \big\|^2_{\dot{H}^1} =: m_k > 0,$$

which is exactly the squared $\dot{H}^1$-norm of the gradient. A step σ is accepted when the realized gain is a controlled fraction of this prediction,

$$\ell\, \sigma\, m_k \ \le \ J(\varphi^{(k)} + \sigma p) - J(\varphi^{(k)}) \ \le \ u\, \sigma\, m_k, \qquad 0 < \ell < u < 1,$$

the lower (Armijo) bound discarding steps that gain too little and the upper (Goldstein) bound discarding overshoot. We enlarge σ when the realized gain exceeds $u\,\sigma m_k$ and shrink it when the gain falls below $\ell\,\sigma m_k$. The same rule governs the I-ascent with $m_k = \|\nabla_{\dot{H}^1} I(\psi^{(k+1/2)})\|^2_{\dot{H}^1}$.

3.5 Stochastic method

Stochastic gradient ascent replaces the full dual gradients (3.3.1) and (3.3.2) by a random unbiased estimator obtained from sampled entries (or mini-batches) $I^{(k)} \sim \mathrm{Unif}\{1,\dots,n\}$ and $J^{(k)} \sim \mathrm{Unif}\{1,\dots,m\}$. Thus instead of evaluating the full sums of partial derivatives $\mathcal{G}_{\varepsilon,\varphi,i}$ and $\mathcal{G}_{\varepsilon,\psi,j}$ one updates the dual potentials using sampled approximations $\widetilde{\mathcal{G}}^{(k)}_{\varepsilon,\varphi,i}$ and $\widetilde{\mathcal{G}}^{(k)}_{\varepsilon,\psi,j}$:

$$\widetilde{\mathcal{G}}^{(k)}_{\varepsilon,\varphi,i} = a_i - m \exp\left\{\frac{\varphi_i^{(k)} + \psi_{J^{(k)}}^{(k)} - C_{iJ^{(k)}}}{\varepsilon}\right\}, \qquad \widetilde{\mathcal{G}}^{(k)}_{\varepsilon,\psi,j} = b_j - n \exp\left\{\frac{\varphi_{I^{(k)}}^{(k)} + \psi_j^{(k)} - C_{I^{(k)}j}}{\varepsilon}\right\},$$

then

$$\mathbb{E}\left[\widetilde{\mathcal{G}}^{(k)}_{\varepsilon,\varphi,i}\right] = \mathcal{G}^{(k)}_{\varepsilon,\varphi,i}, \qquad \mathbb{E}\left[\widetilde{\mathcal{G}}^{(k)}_{\varepsilon,\psi,j}\right] = \mathcal{G}^{(k)}_{\varepsilon,\psi,j}.$$

To reduce the variance in comparison to the single-sample updates one can consider the mini-batch version. Take $B_j^{(k)} \subset [m]$ and $B_i^{(k)} \subset [n]$ and consider the following approximation

$$\widetilde{\mathcal{G}}^{(k)}_{\varepsilon,\varphi,i} = a_i - \frac{m}{|B_j^{(k)}|} \sum_{j \in B_j^{(k)}} \exp\left\{\frac{\varphi_i^{(k)} + \psi_j^{(k)} - C_{ij}}{\varepsilon}\right\},$$

$$\widetilde{\mathcal{G}}^{(k)}_{\varepsilon,\psi,j} = b_j - \frac{n}{|B_i^{(k)}|} \sum_{i \in B_i^{(k)}} \exp\left\{\frac{\varphi_i^{(k)} + \psi_j^{(k)} - C_{ij}}{\varepsilon}\right\}.$$

Then the potential updates follow

$$\varphi_i^{(k+1)} = \varphi_i^{(k)} + \eta\, \widetilde{\mathcal{G}}^{(k)}_{\varepsilon,\varphi,i}, \quad \forall i \in [n], \quad \text{and} \quad \psi_j^{(k+1)} = \psi_j^{(k)} + \eta\, \widetilde{\mathcal{G}}^{(k)}_{\varepsilon,\psi,j}, \quad \forall j \in [m]. \tag{3.5.1}$$

Remark 3.5.1. A common practice in optimization is to use Nesterov acceleration to improve the basic gradient method's sublinear rate of convergence from $O(1/K)$ to $O(1/K^2)$ in the convex smooth case (Nesterov, 2004). One chooses to extrapolate the point $\varphi_i^{(k)}$ with the momentum coefficient $\beta \in [0, 1)$

$$\widetilde{\varphi}^{(k)} = \varphi^{(k)} + \beta\left(\varphi^{(k)} - \varphi^{(k-1)}\right),$$

followed by the update

$$\varphi^{(k+1)} = \widetilde{\varphi}^{(k)} + \eta\, \mathcal{G}^{(k)}_{\varepsilon,\widetilde{\varphi}}.$$

Chapter summary

This closes the review part of the thesis. Five solvers were derived, each from its own formulation. The simplex method solves the discrete problem (3.0.3) exactly. The Sinkhorn algorithm maximizes the entropic dual (3.0.5), the gradient methods, from plain ascent to Adam, maximize the same dual directly, and the stochastic method replaces their full gradients with sampled ones. The Back-and-Forth method maximizes the semi-dual by alternating Sobolev gradient ascent steps in the two potentials with c-transforms between them. Every solver returns a transport plan, except Back-and-Forth, which returns a map. This difference decides which comparisons are meaningful. Part 4 implements all five in one framework and compares them on one testbed for accuracy, runtime and stability.

Part 4

Comparison Study

This part provides a comparative analysis of the performance of different OT methods. We benchmark the numerical methods for discrete OT under the squared Euclidean cost, first in the two-marginal setting and then in the multi-marginal setting. As seen in practice, it may be hard to obtain an ideal solver, which suggests to study accuracy-stability-efficiency trade-offs across dimension, problem sizes and distributions. Throughout, accuracy refers to the quality of the transport cost approximation relative to the exact solver, stability to whether a run produces finite iterates and reaches its stopping criterion within the iteration limit, and efficiency to the runtime needed to do so. Section 4.3 makes these notions precise. As for distributions, two origins of data are considered: synthetic as a probability distributions in one, two and three dimensions, and real-world ones (e.g., images). The solvers include linear programming, Sinkhorn variants, gradient ascent variants, L-BFGS, and the Back-and-Forth method, all specialized to the squared Euclidean cost. Real-data validation is done via colour transfer.

The experiments were performed on compute nodes equipped with two Intel Xeon Gold 6448Y processors and 256 GiB of system memory. GPU-accelerated runs were carried out on an NVIDIA H100 NVL GPU with 94 GB of high-bandwidth memory. All implementations were written in Python 3.13, using the JAX library (version 0.6) for GPU acceleration via the XLA compilation backend. The full benchmarking framework, `uot-bench`, is publicly available.[3]

4.1 Experimental setup

We set the dimension $d \in \{1, 2, 3\}$. The discrete case relies on the discrete representation of the domain, which in the case of OT can be represented in terms of computations in two main structures: *point cloud* and *grid*. In the point cloud case, the support points $\{x_i\}_{i=1}^n$ and $\{y_j\}_{j=1}^m$ are sampled with respect to the underlying continuous distributions μ and ν respectively. This way the point cloud representation does not differ from the discretization introduced in (3.0.1). In contrast, the grid discretization employs a strict rule. We choose a regular lattice on the bounded domain $\Omega = [-1, 1]^d$ in the dimension d. This way, with the number of the discretization points n per axis, the total number of points per each marginal is n^d.

Table 1 lists the discretization sizes used for each dimension.

$d = 1$	$n, m \in \{32, 64, 128, 256, 512, 1024, 2048\}$
$d = 2$	$n \in \{32, 64, 128, 160, 192\}$
$d = 3$	$n \in \{16, 24, 32\}$

Table 1
Discretization sizes for the marginals on the synthetic datasets.

The sampled laws are supported on the finite set $\mathcal{X} \subset \Omega$, so all their moments are finite and Wasserstein costs with polynomial ground metrics are well defined and numerically stable. This holds even for families whose continuous versions have no finite moments, such as the Cauchy distribution below.

Across all experiments we employ the squared Euclidean cost $c(x, y) = \|x - y\|^2$, which is the canonical choice in OT (Brenier, 1991; Santambrogio, 2015) and essential for map-based solvers such as the Back-and-Forth algorithm.

Regularization parameters are varied across several orders of magnitude, $\varepsilon \in \{10^{-3}, 10^{-2}, 10^{-1}, 1\}$.

[3] `https://github.com/AI-Quantum-Research-Group/uot-bench`

Stopping criteria are unified across all solvers to ensure fair comparison, where it is possible. Specifically, we terminate iterations once either of the following conditions are satisfied:

$$\max\left\{\|\pi\mathbf{1}-a\|_2,\ \|\pi^\top\mathbf{1}-b\|_2\right\}\le\tau_{\text{feas}}\quad\text{or we hit the maximum number of iterations.}$$

Here τ_{feas} controls marginal feasibility: set to 10^{-6} for entropic solvers and 10^{-8} for LP. The case of the stopping criteria for the Back-and-Forth method will be discussed separately (see Section 4.3.1).

We consider a collection of probability distributions with varying parameters designed to span different challenges for OT methods: modality, tail thickness, skewness, and anisotropy (i.e., direction-dependent variance/covariance).

(a) **Multivariate Exponential**. This serves as a representative of *light-tailed* families, with density on the d-dimensional space for $x\in\mathcal{X}\subset\mathbb{R}^d$ and independent components given by

$$f_{\text{Exp}}(x;\theta)=\prod_{i=1}^{d}\theta_i\,\exp\{-\theta_i\,x_i\}\quad\text{for } x_i\ge 0,\ \theta=(\theta_1,\dots,\theta_d)^\top\in\mathbb{R}^d_+.$$

Throughout this list, $f(x;\theta)$ denotes the density evaluated at the point x, with the parameters of the family collected after the semicolon. We allow both near-degenerate cases and cases with light-tailed behaviour, producing anisotropic but uncorrelated marginals. Accordingly, we sample the rates as $\theta_i\sim\text{Unif}[0.1,\frac{1}{2}\max_{j\in[n]}x_j]$. The lower bound keeps the rate away from the degenerate flat regime, and the upper bound keeps the decay length $1/\theta_i$ of the order of the domain size, so the sampled density does not collapse into a spike at the origin.

(b) **Student's t Distribution**. The Student's t family interpolates between heavy- and light-tailed behaviour via the *degrees of freedom* parameter $\nu>0$. Its density is

$$f_{\text{St.}\,t}(x;\mu,\Sigma,\nu)=\frac{\Gamma(\frac{\nu+d}{2})}{\Gamma(\frac{\nu}{2})(\nu\pi)^{d/2}|\Sigma|^{1/2}}\left[1+\frac{1}{\nu}(x-\mu)^\top\Sigma^{-1}(x-\mu)\right]^{-\frac{\nu+d}{2}}$$

for $x\in\mathcal{X}\subset\mathbb{R}^d$, with location parameter $\mu\in\mathbb{R}^d$, symmetric positive-definite scale matrix $\Sigma\in\mathbb{R}^{d\times d}$, and $\nu>0$.

When $\nu=1$, the Student's t reduces to the multivariate Cauchy distribution[4]:

$$f_{\text{Cauchy}}(x;\mu,\Sigma)=\frac{1}{\pi^d|\Sigma|^{1/2}}\left[1+(x-\mu)^\top\Sigma^{-1}(x-\mu)\right]^{-(d+1)/2}$$

For $\nu\to\infty$ the density converges pointwise to the Gaussian density, and the Student's t distribution is itself a Gaussian scale mixture (McNeil et al., 2005).

We consider $\nu\in\{2,4,6\}$, which yields heavy-tailed behaviour, with $\nu=4$ and $\nu=6$ closer to the Gaussian limit above, since the moments of order below ν are finite and the tails thin out as ν grows. The Cauchy case ($\nu=1$) is treated separately. We set $\mu\sim\text{Unif}(\mathcal{X})$ and take Σ diagonal, so the marginals are uncorrelated and the difficulty lies in the tails, not in the anisotropy.

[4]The reduction follows from the gamma function identities $\Gamma(1)=1$ and $\Gamma(1/2)=\sqrt{\pi}$ and direct substitution.

(c) **Gaussian Families**. Multivariate normal distributions are fundamental in probability and statistics. The probability density function f_N of the multivariate Gaussian distribution with the parameters $\mu_k \in \mathbb{R}^d$ and positive definite $\Sigma_k \in \mathbb{R}^{d\times d}$ reads as

$$f_N(x\,;\mu_k,\Sigma_k) = \frac{1}{(2\pi)^{\frac{d}{2}}\,|\Sigma_k|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)\right].$$

Finite mixtures of $M \in \mathbb{N}$ such components are formed as weighted sums with weights $\omega = (\omega_1,\ldots,\omega_M) \in [0,1]^M$, $\sum_{k=1}^M \omega_k = 1$:

$$f_{\mathrm{GMM}}(x\,;\mu,\Sigma) = \sum_{k=1}^{M} \omega_k f_N(x\,;\mu_k,\Sigma_k).$$

We sample the density with $M \in \{1,2,4,6\}$ components with $\mu_k \sim \mathrm{Unif}(\mathcal{X})$ and Σ_k drawn from the Wishart prior with diagonal-dominant scale to favour variance over covariance.

(d) **Generalized Hyperbolic Mixtures**. We give the *Generalized Hyperbolic* distribution in the Prause parametrization (Prause, 1999), which is widely adopted in the software packages, for example, by the SciPy Python library (Virtanen et al., 2020). The probability density function in the Prause parametrization is given by

$$\begin{aligned} f_{\mathrm{GH}}(x\,;\mu,\delta,\alpha,\beta,\lambda) = {} & \frac{(\alpha^2-\beta^2)^{\lambda/2}}{\sqrt{2\pi}\,\alpha^{\lambda-1/2}\,\delta^\lambda\,K_\lambda\big(\delta\sqrt{\alpha^2-\beta^2}\big)}\ \exp\{\beta(x-\mu)\} \\ & \times \frac{K_{\lambda-1/2}\big(\alpha\sqrt{\delta^2+(x-\mu)^2}\big)}{\Big[\sqrt{\delta^2+(x-\mu)^2}\,/\,\alpha\Big]^{\frac{1}{2}-\lambda}}, \end{aligned} \tag{4.1.1}$$

where $K_\nu(\cdot)$ is the modified Bessel function of the third kind (Abramowitz and Stegun, 1964, Section 9.6). The parameter μ is the location parameter, $\delta > 0$ is the scale parameter, $\alpha > 0$ is the tail parameter, β is the skewness parameter with $|\beta| < \alpha$, λ is the index (shape) parameter. Formula (4.1.1) is the univariate density. In dimension d we build each mixture component as a product of d independent univariate GH densities, with the location sampled per coordinate and the parameters $\lambda, \alpha, \beta, \delta$ shared within the component. The univariate densities are evaluated with `scipy.stats.genhyperbolic`. The Generalized hyperbolic distribution is extensively used in finance due to the ability to capture asymmetry, heavy tails and volatility clustering (Prause, 1999; Barndorff-Nielsen and Halgreen, 1977; McNeil et al., 2005; Browne and McNicholas, 2015). We set the parameters to be $\lambda \sim \mathrm{Unif}[-2,2]$ (the index selecting the subfamily, e.g., $\lambda = 1$ gives the hyperbolic distribution and $\lambda = -1/2$ the normal-inverse Gaussian), $\alpha \sim \mathrm{Unif}[1,5]$, $\beta \sim \mathrm{Unif}[-0.9\,\alpha,\, 0.9\,\alpha]$, $\delta \sim \mathrm{Unif}[0.05, 0.3]$, $M \in \{2,4,6\}$ components and $\mu \sim \mathrm{Unif}(\mathcal{X})$. These ranges cover varying tail thickness and kurtosis.

Overall, this collection of distributions and parameters provides an informative benchmark for OT. By adjusting tail thickness, we vary the mass spread and hence the transport distance scales. By altering skewness, we introduce asymmetric challenges. By changing modality, we test whether solvers correctly distribute mass across separated modes. Finally, by varying anisotropy, we probe numerical stability, convergence, and the accuracy of dual potentials.

4.2 Solver families and implementation details

We use the simplex method as the exact, unregularized baseline against which the regularized and first-order solvers are measured, since it solves the discrete problem (3.0.3) to optimality and its output therefore serves as the ground truth. We take the implemented solver from the POT (Python Optimal Transport) library (Flamary et al., 2021). Specifically, we use the core routine `ot.lp.emd`, which formulates the discrete OT problem as a linear program. The routine solves this linear program with the network-simplex algorithm bundled with the library. In practice, besides stopping at an optimal vertex of the polytope, the routine caps the number of iterations, and we set that cap to 10^5.

L-BFGS. For the dual problem (3.0.5), we use the L-BFGS optimizer (second-order gradient-based method) from the `jaxopt` library. The solver is run with prescribed regularization parameter ε, tolerance 10^{-6}, maximum number of iterations of $100,000$ and line-search limit of 100 iterations.

Primal–Dual Hybrid Gradient (PDHG). We use the implementation of the primal-dual hybrid algorithm with **chi-square regularization** implemented by Ruban and Gerolin (2025), a member of the accelerated primal–dual family studied for OT and barycenter problems by Chambolle and Contreras (2022), which is applied to the saddle-point formulation

$$\min_{\pi\geq 0}\ \max_{\varphi,\psi} \sum_{i=1}^{n}\sum_{j=1}^{m} \pi_{ij}\, C_{ij} + \frac{\varepsilon}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} \frac{\pi_{ij}^2}{a_i\, b_j} + \sum_{i=1}^{n} \varphi_i\,(a-\pi \mathbf{1}_m)_i + \sum_{j=1}^{m} \psi_j\,(b-\pi^\top \mathbf{1}_n)_j\,. \tag{4.2.1}$$

To state the iterations, it is convenient to work with the density of the plan π relative to $\mu\otimes\nu$, that $\widetilde{\pi}_{ij} = \pi_{ij}/(a_i\, b_j)$. Then for step sizes $\tau,\sigma>0$, the PDHG iterations read

$$\widetilde{\pi}_{ij}^{(k+1)} = \left(\frac{\widetilde{\pi}^{(k)} - \tau(C_{ij} - \varphi_i^{(k)} - \psi_j^{(k)})}{1+\tau\varepsilon}\right)_+, \tag{4.2.2}$$

$$\varphi_i^{(k+1)} = \varphi_i^{(k)} + \sigma\left(1 - \sum_{j=1}^{m} b_j\,\left(2\widetilde{\pi}_{ij}^{(k+1)} - \widetilde{\pi}_{ij}^{(k)}\right)\right), \qquad i\in[n], \tag{4.2.3}$$

$$\psi_j^{(k+1)} = \psi_j^{(k)} + \sigma\left(1 - \sum_{i=1}^{n} a_i\,\left(2\widetilde{\pi}_{ij}^{(k+1)} - \widetilde{\pi}_{ij}^{(k)}\right)\right), \qquad j\in[m]. \tag{4.2.4}$$

The iterations are terminated once the marginal residual falls below a prescribed tolerance:

$$\left\|\pi^{(k)}\mathbf{1}_m - a\right\|_2 + \left\|(\pi^{(k)})^\top \mathbf{1}_n - b\right\|_2 \ \leq \mathrm{tol}\,.$$

In practice, we use the adaptive restarted PDHG implementation of Ruban and Gerolin, which combines the primal–dual above with adaptive stepsize and restart strategies for improved numerical stability and faster convergence (Ruban and Gerolin, 2025).

Sinkhorn Family. The Sinkhorn algorithm admits several practical versions and stabilization schemes, summarized below:

1. The **standard Sinkhorn** iterations are often written in terms of scaling vectors $u\in\mathbb{R}^n$ and

$v \in \mathbb{R}^m$, which are related to the dual Kantorovich potentials by

$$u_i = \exp\left(\frac{\varphi_i}{\varepsilon}\right), \qquad v_j = \exp\left(\frac{\psi_j}{\varepsilon}\right).$$

With the kernel matrix $K_{ij} = \exp(-C_{ij}/\varepsilon)$, the regularized transport plan admits the factorization

$$\pi^\varepsilon = \mathrm{diag}(u)\, K\, \mathrm{diag}(v),$$

and thus the updates (3.2.1) in terms of scalings u and v are viewed as

$$u^{(k+1)} = \frac{a}{Kv^{(k)}}, \qquad v^{(k+1)} = \frac{b}{K^\top u^{(k+1)}}, \tag{4.2.5}$$

where the division is taken componentwise.

2. **Log-Domain Sinkhorn**, which avoids numerical underflow when ε is small by working in logarithmic variables $\mathrm{lnu} = \log u$ and $\mathrm{lnv} = \log v$. Using the *log-sum-exp trick*, the updates become:

$$\begin{cases} \mathrm{lnu}_i^{(t+1)} = \log \mu_i - \mathrm{LSE}_j\left(\frac{g_j^{(t)} - C_{ij}}{\varepsilon}\right), \\ \mathrm{lnv}_j^{(t+1)} = \log \nu_j - \mathrm{LSE}_i\left(\frac{f_i^{(t+1)} - C_{ij}}{\varepsilon}\right), \end{cases}$$

where the operator $\mathrm{LSE}_k(z_k) = \log\left(\sum_k e^{z_k}\right)$ is computed in a numerically stable way as

$$\mathrm{LSE}_k(z_k) = z_{\max} + \log\left(\sum_k e^{z_k - z_{\max}}\right),$$

where $z_{\max} = \max z_k$. This "log-sum-exp trick" ensures that exponentials of large negative values do not underflow to zero and maintains stability for small ε.

Both of the variants are run with the maximum number of iteration of 10^6.

Gradient Ascent Variants. For the numerical experiments, we consider three first-order variants of the gradient ascent.

1. Plain gradient ascent with updates (3.3.3)-(3.3.4), constant stepsize $\eta = 0.001$, and maximal number of iterations 10^6.
2. SGD with momentum, with updates (3.5.1), stepsize $\eta = 0.002$, momentum coefficient $\beta = 0.9$, and maximal number of iterations 10^6.
3. Adam-based gradient ascent, with updates (3.3.6), parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\kappa = 10^{-8}$, stepsize $\eta = 0.001$, and maximal number of iterations 10^6.

Back-and-Forth. We generalize the implementation of the method introduced in Jacobs and Léger (2020) to an arbitrary dimension $d \geq 1$. We make the following implementation choices:

1. We use the *cell-centered* discretization of the domain $\Omega = [0,1]^d$ into n cells with the step $h = (1-0)/N$

$$\left\{x_i = 0 + \frac{1 + 2(i-1)}{2} h : i \in [n]\right\}. \tag{4.2.6}$$

2. Following Jacobs and Léger (2020), the initial step size is chosen based on the maximum of the two measures μ and ν:

$$\sigma^{(0)} = \sigma_s \min\left\{\|\mu\|_\infty^{-1}, \|\nu\|_\infty^{-1}\right\} = \sigma_s / \max\left\{\|\mu\|_\infty, \|\nu\|_\infty\right\},$$

where $\sigma_s > 0$ is an adjusting constant. Whereas Jacobs and Léger (2020) sets $\sigma_s = 8$, we allow different values and provide additional analysis of the performance of this method for the considered values of σ_s.

3. The c-transform can be expressed in terms of the Legendre transform $\varphi^*(y)$ of φ:

$$\varphi^*(y) = \sup_x x \cdot y - \gamma(x), \quad \text{with } \gamma(x) = \frac{1}{2}|x|^2 - \varphi(x). \tag{4.2.7}$$

In the discrete setting, where we have $\{x_i\}_{i=1}^n$ with $x_1 < x_2 < \cdots < x_n$, this becomes

$$\varphi^*(y) = \max_{x_i \in X} L_i(y), \qquad L_i(y) := x_i y - \gamma(x).$$

Since the points x_i are ordered, the Legendre transform φ^* can be evaluated using the Convex Hull Trick (the standard method for computing the upper envelope of affine functions). The pseudocode for the Convex Hull Trick is presented in Appendix B.1.

4. The pushforward $T_\# \rho$ is approximated either by a multilinear (Cloud-in-Cell, CIC) remapping or by an adaptive sampling-based remapping. Full implementation details are provided in Appendices B.2 and B.3.

5. Throughout the numerical experiments, we use the relative $\dot{H}^1$-norm change in ψ as the stopping criterion, terminating once it falls below 10^{-4}.

TABLE 2
Error metrics available in the implementation of the Back-and-Forth method.

Error metric	**Definition**
TV error on the ψ-side	$0.5 \sum_{x \in \mathcal{X}} \left\lvert T_{\varphi^{(k)}\#}\mu(x) - \nu(x) \right\rvert$
L_∞ error on the ψ-side	$\max_{x \in \mathcal{X}} \left\lvert T_{\varphi^{(k)}\#}\mu(x) - \nu(x) \right\rvert$
$\dot{H}^1$ seminorm of the ψ-update	$\|\nabla_{\dot{H}^1} I(\psi^{(k)})\|_{\dot{H}^1} = \left\langle T_{\varphi^{(k)}\#}\nu - \mu, (-\Delta)^{-1}(T_{\varphi^{(k)}\#}\nu - \mu)\right\rangle_{\dot{H}^{-1}}$
Relative $\dot{H}^1$-norm change in ψ	$\frac{\left\lvert \|\nabla_{\dot{H}^1} I(\psi^{(k)})\|_{\dot{H}^1} - \|\nabla_{\dot{H}^1} I(\psi^{(k-1)})\|_{\dot{H}^1} \right\rvert}{\|\nabla_{\dot{H}^1} I(\psi^{(k)})\|_{\dot{H}^1}}$
Absolute change in the dual objective	$\lvert \mathcal{D}(\varphi_n, \psi_n) - \mathcal{D}(\varphi_{n-1}, \psi_{n-1}) \rvert$
Relative change in the dual objective	$\frac{\left\lvert \mathcal{D}(\varphi^{(k)}, \psi^{(k)}) - \mathcal{D}(\varphi^{(k-1)}, \psi^{(k-1)}) \right\rvert}{\lvert \mathcal{D}(\varphi^{(k)}, \psi^{(k)}) \rvert}$

4.3 Descriptive analysis

Stability. We distinguish between two notions of stability: numerical stability and convergence stability. Numerical stability refers to whether the solver produces finite, well-defined iterates (no `NaN`/overflow). Convergence stability is measured by whether the solver reaches the stopping criterion before hitting the iteration limit.

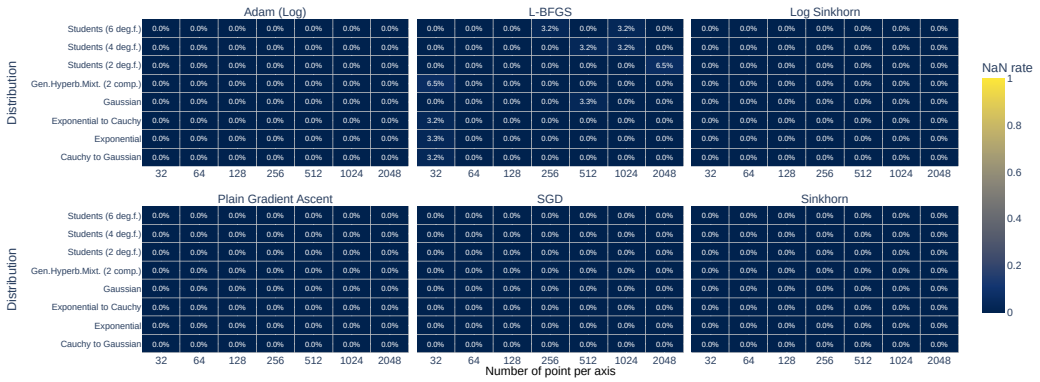


FIGURE 1. *`NaN` failure rate (numerical instability) for solvers under $\varepsilon = 0.001$ across 1D distributions and grid resolutions (only those with failures are shown). No solver produces `NaN` iterates on these instances. L-BFGS shows a few isolated failures, at most one run per cell.*

FIGURE 2. *Maximum-iteration hit rates at $\varepsilon = 0.001$ across 1D distributions and grid resolutions. The figure shows a clear difference in convergence stability of the methods. Log-domain Sinkhorn remains stable throughout the tested settings. Standard Sinkhorn does not, and reaches the iteration limit on most runs. Plain gradient ascent stalls in most cases. L-BFGS is generally reliable on smaller grids but loses stability on the finest discretizations, while Adam and SGD also show convergence difficulties on several of the more challenging distributions.*

Figure 1 shows that no solver produces `NaN` failures on these instances, apart from a few isolated L-BFGS runs. Figure 2 and Figure 3 address convergence stability. Log-domain Sinkhorn

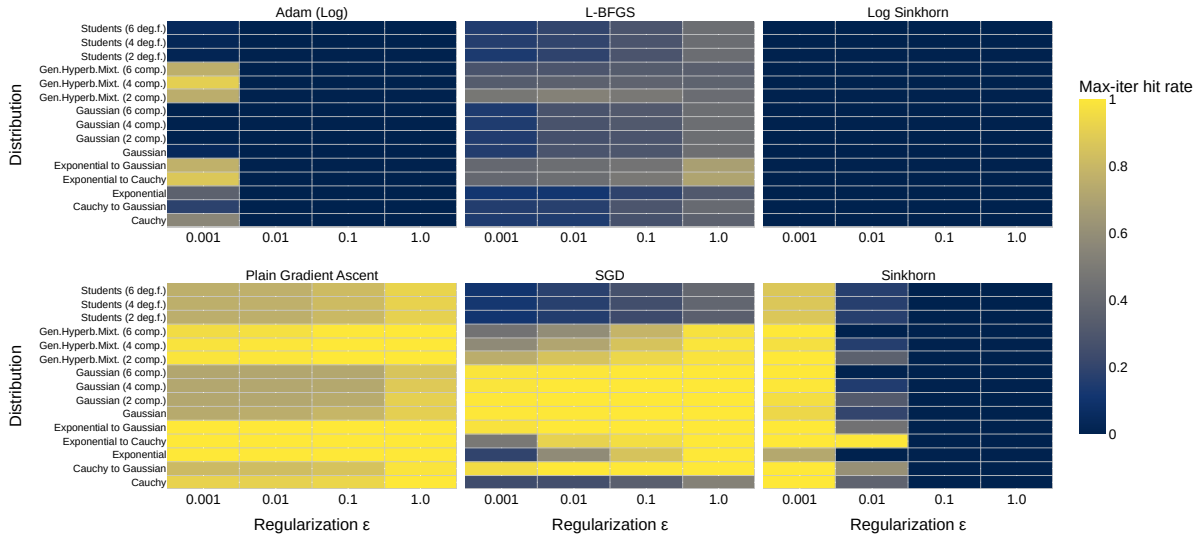


FIGURE 3. *Maximum-iteration hit rates as a function of the regularization parameter* $\varepsilon \in \{0.001, 0.01, 0.1, 1.0\}$ *across 1D distributions. The main trend is that the effect of regularization depends strongly on the solver. Log-domain Sinkhorn remains stable for all values of* ε*, whereas plain gradient ascent stalls almost uniformly. Adam is least stable at the smallest* ε *and improves as* ε *increases. Standard Sinkhorn behaves the same way, and is least stable at the smallest* ε*.*

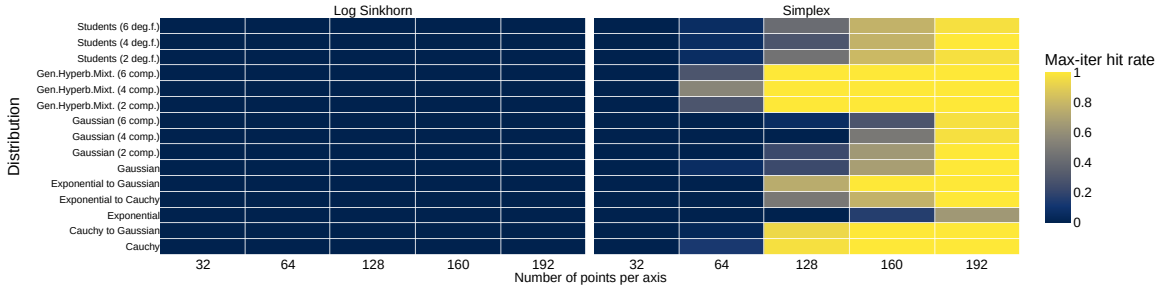


FIGURE 4. *Maximum iteration hit rate across all 2D distributions and grid resolutions (32-192 points per axis) for log-domain Sinkhorn and simplex. Sinkhorn remains stable across all settings, while simplex loses convergence with the refinement of the grid.*

remains stable across grid sizes and regularization values, while standard Sinkhorn, the gradient-based methods and L-BFGS show convergence problems at fine grids or small ε. This is what separates the two Sinkhorn variants. The log-domain form updates the potentials φ and ψ directly through (3.2.1), so it still converges at the smallest ε. The standard form iterates the scalings $u = \exp(\varphi/\varepsilon)$ against the kernel $K = \exp(-C/\varepsilon)$ instead, as in (4.2.5). Dividing by ε inside the exponential is what costs precision. At $\varepsilon = 10^{-3}$ a cost of $C_{ij} = 1$ gives $K_{ij} = e^{-1000}$, which rounds to zero in double precision. The potentials stay comparable in size to the entries of C, so no such rounding occurs. This is why the log-domain form is the recommended one at small ε (Schmitzer, 2019). Simplex and PDHG are not shown in these plots because they converge in all of the 1D

runs.

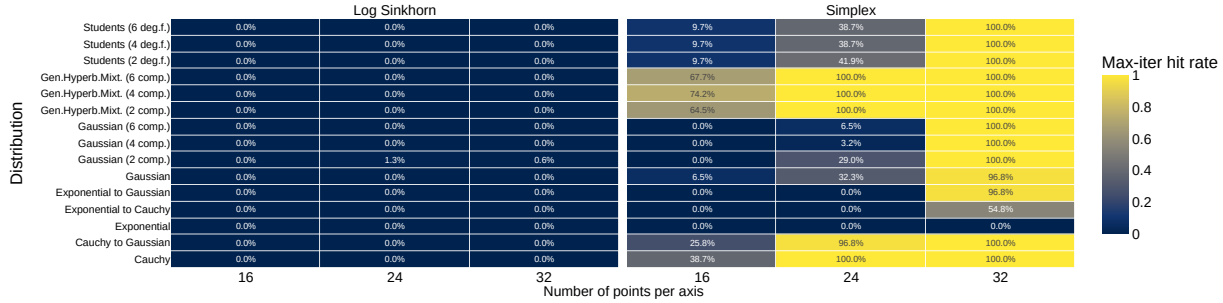


FIGURE 5. *Maximum iteration hit rate across all 3D distributions for log-domain Sinkhorn and simplex. As in 2D, log-domain Sinkhorn remains reliably convergent across the tested settings, while simplex reaches the maximum iteration capacity on a large fraction of runs.*

The higher-dimensional results, in 2D and 3D, lead to the same conclusion (Figures 4 and 5): the log-domain Sinkhorn remains reliably convergent, whereas the simplex method becomes impractical with refinement of the grid. We also do not report the performance of PDHG in 2D because of its memory cost: storing the dual potentials with the full transport plan becomes impractical as the grid grows.

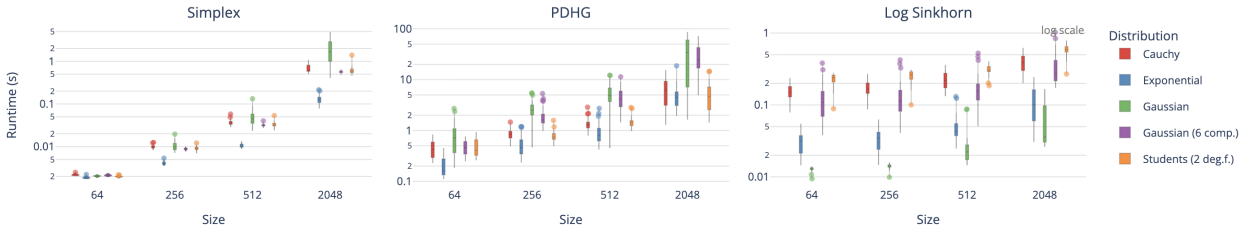


FIGURE 6. *Runtime distributions of simplex, PDHG ($\varepsilon = 20$) and log-domain Sinkhorn ($\varepsilon = 0.001$) on 1D Cauchy, Exponential, Student's, and Gaussian distributions, including a 6 component Gaussian mixture. The main pattern is that log-domain Sinkhorn is the most scalable method, while simplex is competitive on small grids.*

Runtime efficiency. Figure 6 shows a clear difference in the scalability of the methods. Log-domain Sinkhorn remains the most scalable method in this comparison. While simplex is competitive on small grids, its runtime increases quickly as the grid size grows. Turning to Figure 7, we see that gradient-based solvers are not competitive with Sinkhorn in the 1D setting: all of them show higher variability and moderate to high runtimes. Taken together with the stability results, this suggests that these methods are less practical than Sinkhorn in terms of both robustness and runtime. The same behaviour persists in 2D and 3D, where log-domain Sinkhorn attains lower median runtimes than simplex. See Figure 8. Because of instability and/or higher computational cost, we do not continue with L-BFGS, gradient-based solvers and PDHG in 2D and 3D.
An interesting feature of log-domain Sinkhorn is that its iteration count slightly decreases as the problem size grows; see Figure 9. A possible explanation is that finer grids provide smoother approximations of the underlying distributions so that the marginals become easier to balance and require fewer correction steps. Thus, although each iteration becomes more expensive on larger

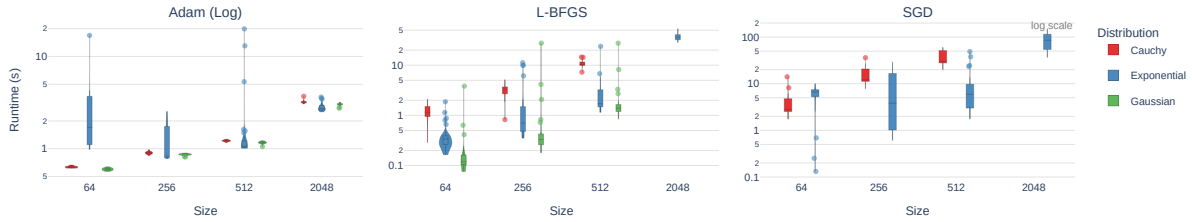


FIGURE 7. *Runtime distributions for gradient-based solvers at $\varepsilon = 0.001$ across Gaussian, Cauchy, and Exponential 1D distributions. L-BFGS shows moderate runtimes but poor scalability, SGD suffers from high variability and consistently large runtimes, while Adam converges more reliably.*

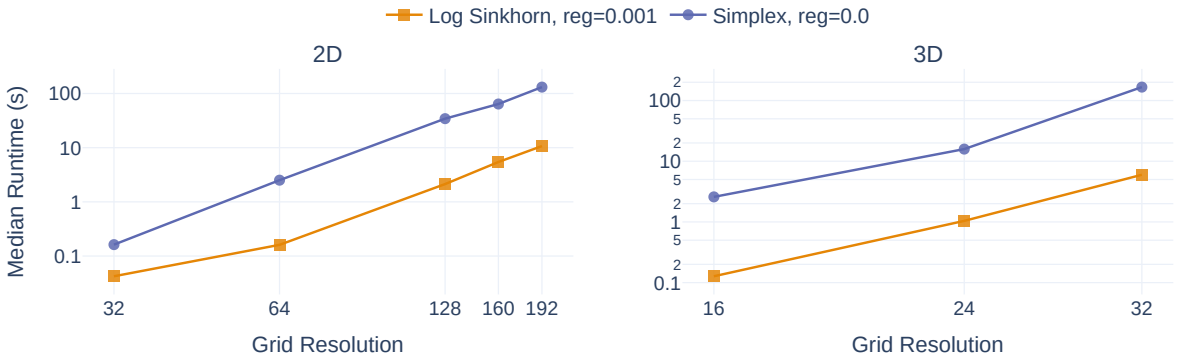


FIGURE 8. *Median runtime scaling of log-domain Sinkhorn ($\varepsilon = 10^{-3}$) and simplex across grid resolutions in higher dimensions ($d = 2, 3$). In both cases, log-domain Sinkhorn attains lower median runtimes than simplex.*

grids, the number of iterations needed to reach the tolerance does not increase and may even decrease slightly. This is consistent with the good runtime behaviour observed in Figure 6.

Efficiency through performance profiles. For the 1D benchmark, where many solvers are compared across many distributions and grid sizes, it is useful to complement the runtime boxplots with a more aggregated view. For this purpose we use the performance profiles of Dolan and Moré (2002).

For each problem instance p and solver s, we define the normalized time ratio

$$t_{p,s} := \frac{T_{p,s}}{\min_{s' \in \mathcal{S}} T_{p,s'}} \geq 1, \qquad (4.3.1)$$

where $T_{p,s}$ is the runtime of solver s on problem instance p. For failed runs we assign a large penalty value $t_F \gg \max_{s' \in \mathcal{S}} t_{p,s'}$.

The performance profile of solver s is the em-

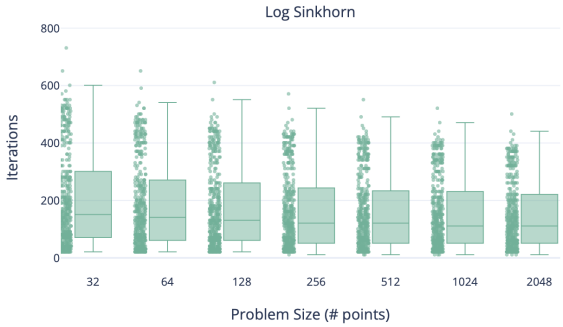


FIGURE 9. *Iteration counts of log-domain Sinkhorn across problem sizes on 1D distributions. The number of iterations does not increase with grid size and in fact decreases slightly. This helps explain why the runtime of log-domain Sinkhorn remains favourable even on finer grids.*

pirical distribution function:

$$P_s(\tau) \;=\; \frac{1}{n_p} \sum_{p \in \mathcal{P}} \mathbf{1}\{\, t_{p,s} \le \tau \,\}, \qquad \tau \ge 1. \tag{4.3.2}$$

Thus, $P_s(\tau)$ is the fraction of instances on which solver s is within a factor τ of the best observed runtime. In particular, $P_s(1)$ gives the fraction of wins, while the right plateau reflects the fraction of instances successfully solved.

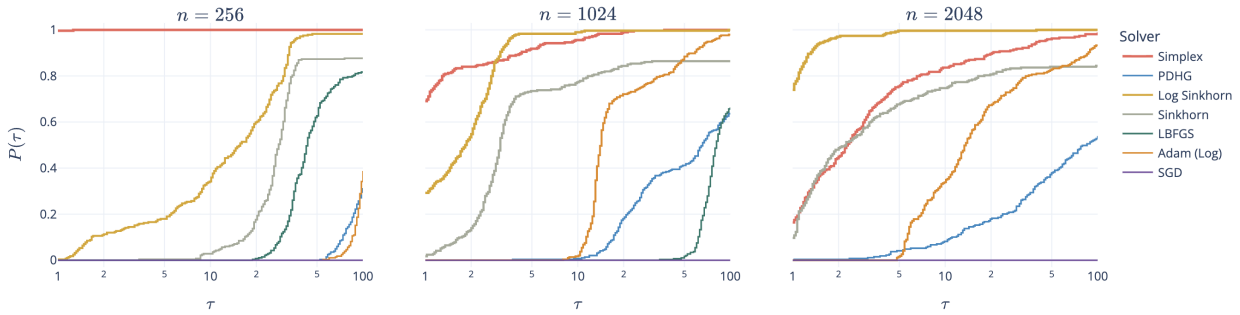


FIGURE 10. *Performance profiles (empirical distribution functions of normalized time ratios) for all 1D problems at grid sizes $n \in \{256, 1024, 2048\}$, with $\varepsilon = 0.001$ for Sinkhorn and $\varepsilon = 10$ for PDHG. Curves that are higher and closer to the left indicate better relative runtime performance, while the right plateau reflects robustness. The figure shows a gradual transition in the best-performing method as the grid is refined: simplex is clearly strongest on the smallest grids, and on the finest grid log-domain Sinkhorn becomes the strongest method overall. PDHG and gradient-based methods remain less competitive.*

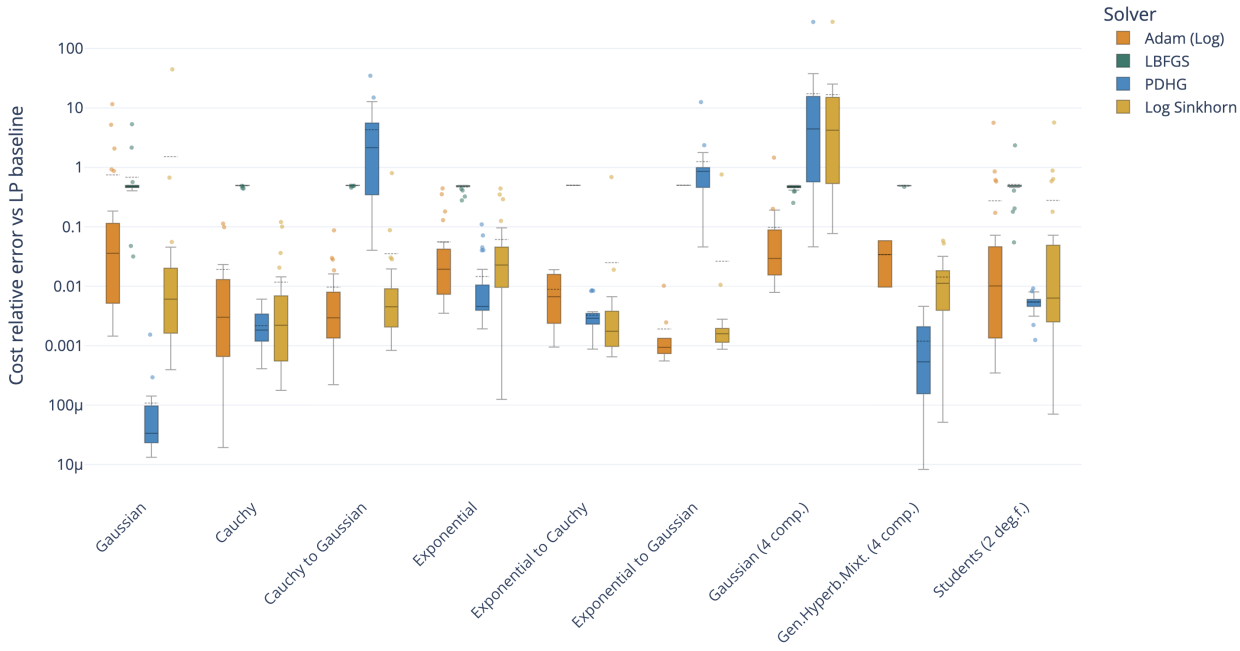
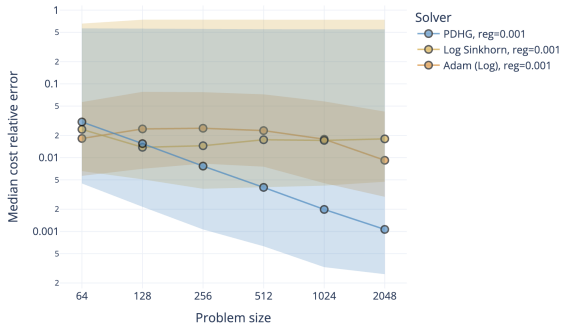


FIGURE 11. *Relative transport cost error (log scale) at grid size $n = 512$ and $\varepsilon = 10^{-3}$ grouped by distribution family. Lower is better. Boxplots summarize variability across instances for Adam (log-domain), L-BFGS, PDHG, and log-domain Sinkhorn.*

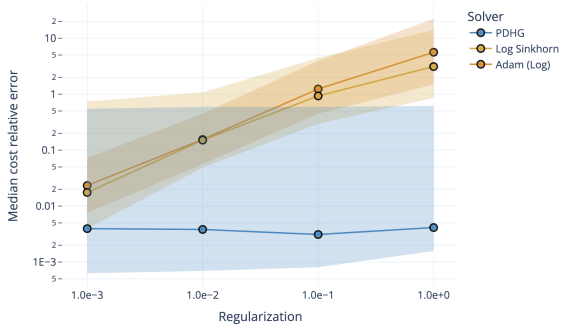
Accuracy. We report accuracy results in 1D, where all six solvers were run. Accuracy in higher dimensions is omitted for the reasons outlined above: specifically, in 2D and 3D, direct accuracy comparison becomes uninformative. The convergence issues of simplex observed in Figure 4 and Figure 5 show that a large fraction of runs hit the iteration cap, so the reference itself becomes unreliable. Higher-dimensional behaviour is therefore covered by the robustness and runtime results.

We assess the accuracy through the quality of the transport cost approximation. Since the low cost value is only meaningful when the marginal constraints are satisfied, we first verify that the solution satisfies those constraints up to a tolerance of 10^{-6}. This allows us to focus on the transport error itself. For the regularized methods, the objective differs from the unregularized ones. To compare all methods in the same physical quantity we evaluate the transport cost of the computed plan π_ε through $\hat{C}_\varepsilon := \langle C, \pi_\varepsilon \rangle$ and, because of different scalings of the problem instances, report the relative error $|\hat{C}_\varepsilon - C^*|/C^*$ where C^* is the reference cost given by the solution of simplex method.

Figure 11 shows that the error behaviour across distribution families is not uniform. The three main methods (Adam, Sinkhorn, PDHG) cannot be clearly ranked from this view: their boxes overlap within most families, and PDHG is not uniformly best. The one clear message is that L-BFGS consistently fails to approximate the cost. To see actual trends, Figure 12 shows the error as a function of problem size and regularization separately. PDHG improves monotonically with the grid size, while Sinkhorn and Adam remain stuck in the 10^{-2} range — this is the regularization bias, which does not shrink with finer discretization at fixed ε. Figure 12b confirms that the error of Sinkhorn and Adam grows near-linearly with ε, while PDHG stays flat. This is expected and reflects a fundamental difference between the methods. Sinkhorn and Adam optimize the entropy-regularized problem, so the recovered cost depends on ε directly. PDHG solves the chi-square regularized OT, which remains stable as $\varepsilon \to 0$ and whose bias approaches the discretization bias in that limit (Ruban and Gerolin, 2025). This is consistent with the PDHG cost error behaviour across the tested range.



(A) *Error as a function of problem size at fixed $\varepsilon = 10^{-3}$. PDHG shows monotone improvement with the grid refinement. Sinkhorn, on the other hand, displays the regularization bias: with fixed regularization parameter ε the refinement of the grid does not improve the accuracy of the method.*



(B) *Error as a function of regularization ε at a fixed grid size $n = 512$. PDHG does not show dependence on the regularization, while Sinkhorn and Adam show a near-linear dependence on ε.*

FIGURE 12. *Median relative transport cost error (vs the LP baseline) for PDHG, log-domain Sinkhorn, and Adam aggregated on 1D problems. The shaded bands show the interquartile range (25-75%). Lower is better.*

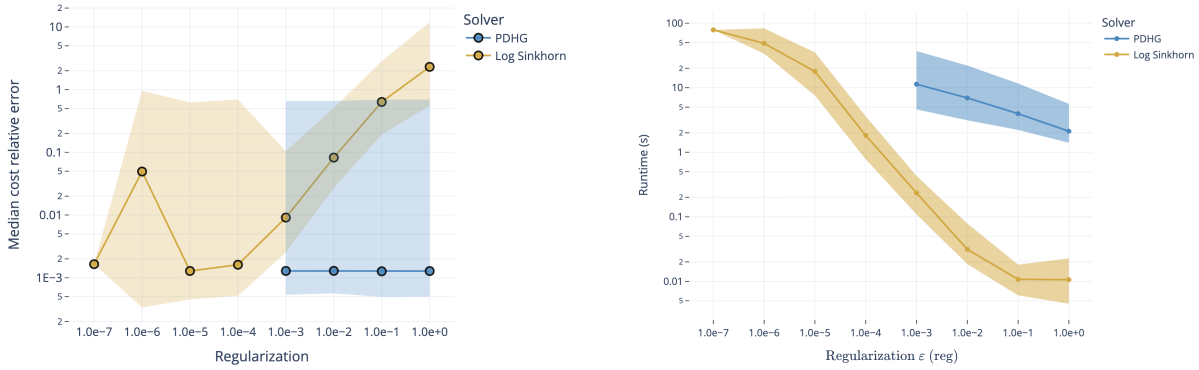
Effect of regularization. The log-sum-exp trick and the stability of log-domain updates in Sinkhorn allow us to consider smaller values of ε. Table 3 reports the maximum iteration hit rate for Sinkhorn (log-domain) on a Gaussian mixture distribution across a wide range of ε. The iteration demand grows once ε drops below the $10^{-5} - 10^{-6}$ range, with the iteration budget set to 10^6 (as noted before).

TABLE 3

Max-iteration hit rates (%) for Sinkhorn (log-domain) solver with Gaussian mixture distribution (6 components, 30 samples) across regularization levels ε. Higher values indicate more frequent reaching of the iteration cap.

Regularization ε	1e−7	1e−6	1e−5	1e−4	1e−3	1e−2	1e−1	1
Max-iter hit rate (%)	**100.0**	**91.0**	4.3	0.0	0.0	0.0	0.0	0.0

As noted earlier, PDHG and Sinkhorn differ in how the regularization enters the objective. In Figure 13 compares both methods on runtime and accuracy across ε, extending the range to small values of ε for Sinkhorn as in Table 3. Note that the observed plateau in runtime and the erratic cost error of Sinkhorn at small ε (Figures 13a, 13b) reflect the method hitting the iteration cap, as confirmed by Table 3. Thus, attaining better precision with Sinkhorn requires decreasing ε and increasing the maximum iteration budget together — consistent with the $O(\log(1/\delta)/\varepsilon)$ iteration count discussed in Section 3.2. More generally, the regularization ε in PDHG acts as an algorithmic parameter, while for Sinkhorn, ε defines the problem itself: smaller ε means a problem closer to the LP.



(A) *Median relative transport cost error (vs the LP baseline) as the function of ε in log-scale. Lower is better.*

(B) *Median runtime as the function of ε. Lower is faster.*

FIGURE 13. *Effect of the regularization ε on accuracy and runtime for log-domain Sinkhorn and PDHG aggregated across all 1D datasets on $n = 2048$ grid. Markers show medians across problem instances and shaded bands show the interquartile range (25-75%)*

The sparsity of the transport plan π — the fraction of zero entries — is a structural property separating the two solver families. Simplex returns a vertex of the feasible polytope, which has at most $n+m-1$ nonzero entries (Section 3.1); the optimal plan $\pi^\star$ is therefore sparse, with $O(n+m)$ active cells out of nm. Sinkhorn, in contrast, returns a fully dense plan: the entropic optimality conditions give $\pi_{ij} = \exp((\varphi_i + \psi_j - C_{ij})/\varepsilon) > 0$ for every (i, j), so no entry can vanish. As $\varepsilon \to 0$ the off-support entries decay (exponentially in $1/\varepsilon$) and the plan becomes *approximately* sparse, recovering the simplex support in the limit without ever attaining exact zeros. Figure 14 shows the Sinkhorn sparsity approaching the simplex baseline as ε decreases.

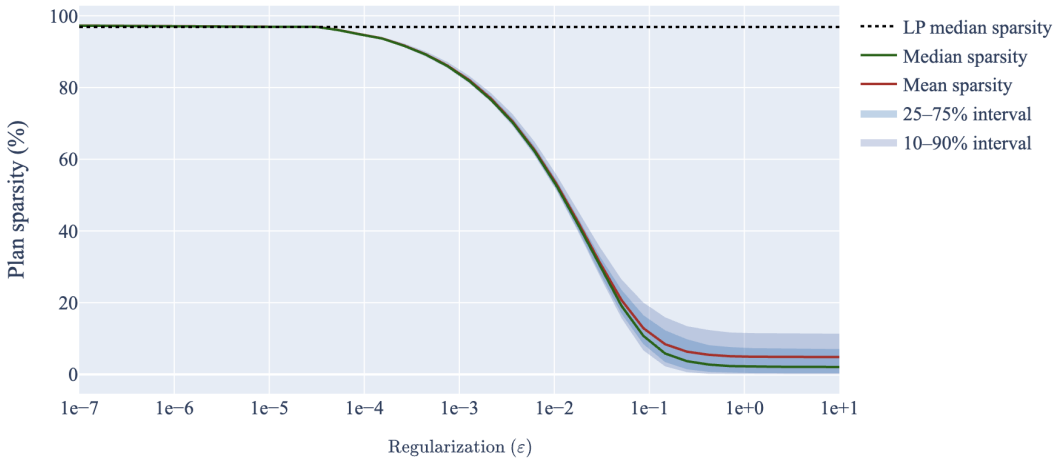


FIGURE 14. *Transport plan sparsity (in %) of the Sinkhorn (log-domain) solver as a function of the regularization parameter ε, with the simplex (LP) median sparsity baseline.*

This is another manifestation of the same principle: as $\varepsilon \to 0$, Sinkhorn does not just approximate the LP cost – it approaches the LP structure, at the price of the increased iteration budget reported in Table 3.

4.3.1 Evaluation of the Back-and-Forth method

Unlike Kantorovich-type solvers that return a transport plan, the Back-and-Forth method constructs a transport map $T : \mathcal{X} \to \mathcal{Y}$ with potentials φ or ψ, addressing the Monge formulation. This suggests benchmarking the Back-and-Forth method against the simplex baseline rather than biased Sinkhorn or PDHG. In this section we assess the stability of the method, how different configurations affect the stability, the runtime, and the accuracy, by utilizing the metrics applicable to the Monge map directly.

Throughout this section transport costs are reported under $c(x, y) = \|x - y\|^2$, consistent with the simplex comparisons of the preceding sections. The Back-and-Forth solver of Jacobs and Léger (2020) is formulated for $\frac{1}{2}\|x-y\|^2$; since the two costs differ only by a global factor, the optimal map is identical and every reported quantity — relative cost error, Monge–Ampère residuals, and convexity — is unchanged.

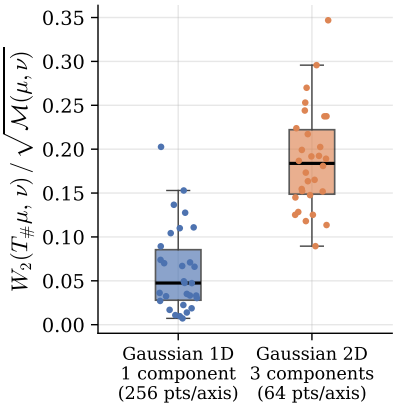


FIGURE 15. *Relative discrepancy $W_2(T_\#\mu, \nu)/\sqrt{\mathcal{M}}$ for the Back-and-Forth pushforward: the numerator (marginal error) is computed with simplex, the denominator with the solver's own cost $\mathcal{M} = \int \|x - T(x)\|^2 \,\mathrm{d}\mu$. Dots are individual instances.*

Stopping criterion. The absolute seminorm $\|\nabla_{\dot{H}^1} I(\psi_n)\|_{\dot{H}^1}$ is dimensional, so its magnitude depends on the scale of each instance and a single fixed threshold is not consistent across our benchmark. We therefore stop on its relative change

$$\frac{\big|\, \|\nabla_{\dot{H}^1} I(\psi_n)\|_{\dot{H}^1} - \|\nabla_{\dot{H}^1} I(\psi_{n-1})\|_{\dot{H}^1} \,\big|}{\|\nabla_{\dot{H}^1} I(\psi_n)\|_{\dot{H}^1}},$$

a dimensionless criterion that transfers across all instances. We validate that it yields converged maps through the marginal error $\tau = W_2(T_\#\mu, \nu)$, reported as the dimensionless ratio $\tau/\sqrt{\mathcal{M}} \ll 1$, where $\mathcal{M} = \int \|x - T(x)\|^2 \,\mathrm{d}\mu$ is the transport cost of the computed map; Figure 15 confirms it stays well below 1 in 1D and 2D, so the $\dot{H}^1$ criterion yields maps consistent with the Wasserstein metric.

TABLE 4
Max-iteration hit rate (%) for representative distribution families in 1D and 2D Back-and-Forth experiments. A(1) and A(8) denote adaptive scheme with initial stepsize $\sigma_s = 1$ and $\sigma_s = 8$; C(1) and C(8) denote CIC.

Dim.	Configuration	A(1)	C(1)	A(8)	C(8)
1D	Gaussian (2 comp.)	20.5	26.2	44.8	36.2
	Gaussian	7.1	26.7	12.9	30.0
	Exponential	12.9	23.8	40.0	32.9
	Cauchy → Gaussian	20.7	21.7	85.7	35.5
2D	Gaussian (2 comp.)	0.7	0.0	3.3	2.7
	Gaussian	0.0	0.0	1.7	0.0
	Exponential	0.0	0.0	0.7	1.3
	Cauchy	3.3	0.7	6.0	8.0

Stability. We assess the stability of each configuration — initial stepsize σ_s and pushforward procedure — through the max-iteration hit rate. Large σ_s already degrades the pushforward visibly, producing ring-shaped artifacts and misplaced mass (Figure 16) and motivating the conservative $\sigma_s = 1$; Table 4 reports the hit rate across representative families. The full tables are given in Appendix D (Table 9, Table 10, and Table 11). The method is markedly more stable in higher dimensions. This matches the iteration counts in Appendix C Figure 31, where 1D runs show oscillatory behaviour and require many iterations (Figure 31a), while 2D runs converge cleanly within 30-35 steps (Figure 31b).

The higher 1D hit rates can plausibly be attributed to discretisation effects: in one dimension

mass can only be redistributed along a single direction, so coarse grid spacing makes local mass imbalances harder to correct. In higher dimensions the same resolution offers more directions along which mass can move, giving the method more flexibility to smooth out discretization artifacts. Consistently with this, heavy tailed Cauchy-type families show larger hit rates than Gaussian and Exponential ones across both dimensions, indicating that tail mass near the domain boundary interacts strongly with the stepsize choice.

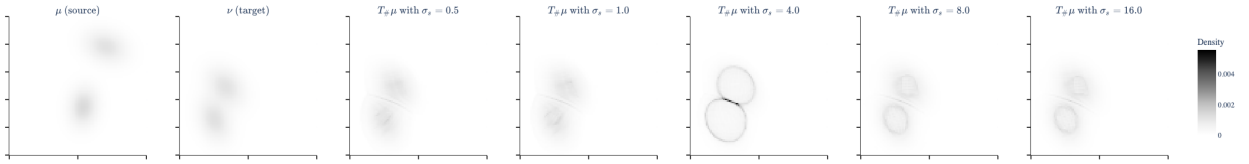


FIGURE 16. *Heatmaps of the pushforward densities in the 2D Gaussian mixture benchmark. From left to right: source density μ, target density ν, and the pushforwards $T_{\#}\mu$ obtained by the Back-and-Forth method for $\sigma_s \in \{0.5, 1, 4, 8, 16\}$. Small stepsizes ($\sigma_s \in \{0.5, 1\}$) reproduce the two-bump structure of ν, while larger ones produce ring-shaped artifacts and misplaced mass, motivating the conservative $\sigma_s = 1$.*

TABLE 5
Pooled total-variation (TV) distance between the numerical pushforward and target. Entries are median [IQR] over converged runs. Bold marks the lowest median TV in each dimension: Adaptive with $\sigma_s = 1$ is best throughout.

Dim.	Scheme	σ_s	Median TV	IQR
1D	CIC	1	0.071	[0.042, 0.122]
	CIC	8	0.091	[0.055, 0.159]
	Adaptive	1	**0.021**	[0.016, 0.030]
	Adaptive	8	0.046	[0.028, 0.077]
2D	CIC	1	0.139	[0.096, 0.196]
	CIC	8	0.160	[0.106, 0.254]
	Adaptive	1	**0.061**	[0.031, 0.081]
	Adaptive	8	0.065	[0.039, 0.125]
3D	CIC	1	0.285	[0.173, 0.355]
	CIC	8	0.280	[0.187, 0.381]
	Adaptive	1	**0.116**	[0.037, 0.178]
	Adaptive	8	0.147	[0.052, 0.277]

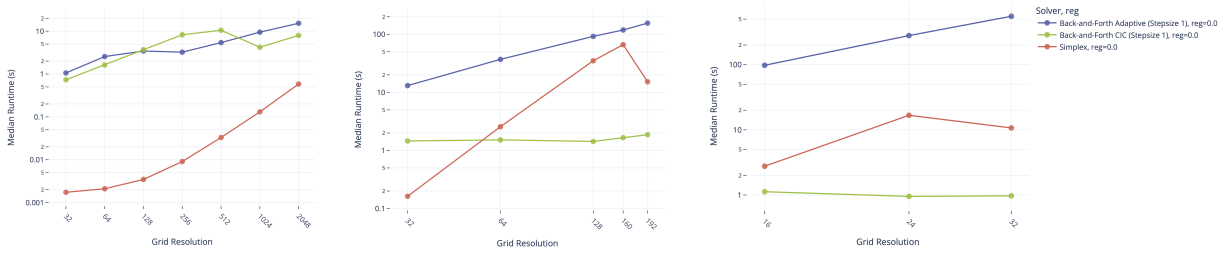
Runtime. Figure 17 compares median runtime against grid resolution: simplex wins in 1D, while the CIC Back-and-Forth scheme scales best in 2D and 3D, where simplex grows steeply and breaks down once the grid is fine.

The per-iteration cost of Back-and-Forth is dominated by two operations: the $c-$transform and the pushforward. The $c-$transform is computed in $O(n)$ per dimension via the Convex Hull Trick (see Section B.1), and the CIC pushforward is a cheap multilinear remapping; the efficiency of the overall method therefore hinges on these two routines.

Accuracy. One natural way to assess the match of marginals under the computed map T is the *Total Variation* (TV) distance

$$\mathrm{TV}(T_{\#}\mu, \nu) = \frac{1}{2} \sum_{j=1}^{m} \big| (T_{\#}\mu)(y_j) - \nu(y_j) \big|, \tag{4.3.3}$$

which vanishes for a perfect Monge solution. This is not an arbitrary divergence: it is the optimal transport distance for the discrete 0–1 cost $c(x, y) = \mathbf{1}_{\{x \neq y\}}$ (Villani, 2009), so it stays inside the OT framework. We prefer it to the Kullback–Leibler divergence, which is not a metric, diverges under the support mismatch of Figure 15. TV is, however, geometry-blind: with the 0–1 cost a small local displacement is penalised as heavily as a gross misplacement, so we use it as a cheap marginal-fidelity diagnostic (Table 5) rather than a primary accuracy metric.



(A) *1D* (B) *2D* (C) *3D*

FIGURE 17. *Median runtime with grid resolution for simplex (red line) and Back-and-Forth (adaptive (blue line) and CIC (green line) pushforward, both $\sigma_s = 1$) aggregated over all instances of a given dimension; vertical axis is logarithmic. Simplex is fastest in 1D but grows steeply and breaks down by 3D, where CIC scheme stays near 1 second.*

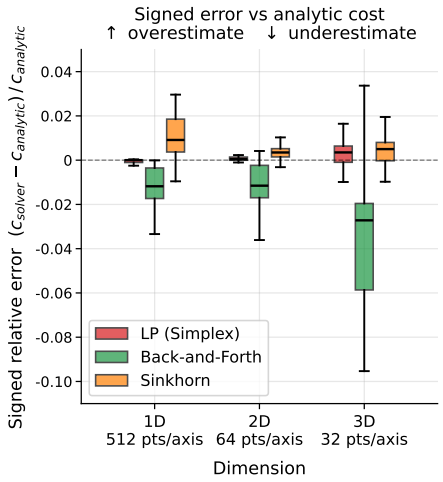


FIGURE 18. *Signed relative cost error $(c_{\text{solver}} - c_{\text{analytic}})/c_{\text{analytic}}$ on Gaussian instances, where c_{analytic} is the closed-form cost (Takatsu, 2011); positive is overestimation, negative underestimation. Grids are 512/64/32 points per axis in 1D/2D/3D. LP (simplex) recovers the analytic cost up to discretization. Back-and-Forth underestimates, increasingly so on the coarser higher-dimensional grids, a discretization effect of the map-based cost (Table 6). Sinkhorn ($\varepsilon = 4 \times 10^{-4}$) overestimates slightly; the offset is its residual entropic bias and shrinks with ε.*

Transportation cost approximation. The Back-and-Forth method reports its cost through a map. This map is different in nature from the barycentric map of Remark A.2: it is returned natively rather than projected from a plan, so the coupling is deterministic and the cost underestimation of that remark does not apply here. The remark governs the plan-derived maps instead, and we return to it in the structural comparison of Figure 19.

The underestimation that Back-and-Forth does exhibit (Figure 18) is a discretization effect of the map-based cost rather than a marginal mismatch. Two observations isolate it. First, the LP cost on the same discretized measures recovers the analytic value, so the binning of μ and ν is not responsible. Second, per instance the cost gap correlates with the marginal error $\tau = W_2(T_{\#}\mu, \nu)$ only on fine grids: strong in 1D at 512 points (Spearman -0.68), it collapses to insignificance (-0.21) when 1D is recomputed at 32 points (Table 6), identifying the dominant effect as grid resolution rather than dimension or marginal mismatch. The gap does not vanish over the tested grid range and grows as the grid coarsens. Sinkhorn ($\varepsilon = 4 \times 10^{-4}$) is included for completeness; its small positive offset is the residual entropic bias of Section 3.2.

Monotonicity of the map. The Brenier map is the gradient of a convex potential, equivalently a monotone map, so its Jacobian ∇T has a positive semi-definite symmetric part. We test this pointwise: at each grid point $x \in \mathcal{X}$ we form $\nabla T(x)$ by central differences of the map, symmetrise it, and check that the smallest eigenvalue of $\text{sym}(\nabla T(x))$ is non-negative up to a tolerance of 10^{-10}. We report the *convexity fraction*, the proportion of points passing this test. It exceeds 0.99 in every configuration and dimension (Table 13), so T is monotone—hence injective, and a bijection onto its image—almost everywhere, which licenses the change of variables used below, even in the most challenging 3D setting.

TABLE 6

Per-instance correlations between the Back-and-Forth cost gap $c_{\mathrm{BFM}} - c_{\mathrm{analytic}}$ and marginal error $\tau = W_2(T_{\#}\mu, \nu)$ for Gaussian instances ($n = 30$ per dimension with closed-form solution (Takatsu, 2011)). The original benchmark uses 512/64/32 points per axis in 1D/2D/3D, whereas the control uses 32 points per axis in every dimension.

Benchmark	Dim.	Pearson	Spearman	OLS slope
Original (512/64/32)	1D	−0.504	−0.675	−0.0035
	2D	−0.061	−0.384	−0.0026
	3D	−0.051	−0.188	−0.0013
Control (32/axis)	1D	0.274	−0.212	0.0184
	2D	0.014	−0.109	0.0003
	3D	−0.167	−0.250	−0.0050

Monge–Ampère residuals. Another useful metric to check the structural correctness of the resulting map is the Monge–Ampère residual check, which certifies the structural properties of the optimal Brenier map. To assess the structural quality we compute the Monge–Ampère residual

$$r(x) = \det(\nabla T(x))\, \rho_\nu(T(x)) - \rho_\mu(x), \tag{4.3.4}$$

and report its L^1 norm as a global measure aggregated over all $x \in \mathcal{X}$.

This L^1 norm measures the misallocated mass. Indeed, define the reweighted target density $\tilde{\rho}(x) := \det(\nabla T(x))\, \rho_\nu(T(x))$. With the change of variables $y = T(x)$, $\mathrm{d}x = \mathrm{d}y / \det(\nabla T(T^{-1}(y)))$ — valid since T is monotone (Table 13), hence a bijection — $\tilde{\rho}$ integrates to one, so both $\tilde{\rho}$ and ρ_μ are probability densities on $\mathcal{X}$ and

$$\tfrac{1}{2}\, \|r\|_{L^1} = \tfrac{1}{2}\, \|\tilde{\rho} - \rho_\mu\|_{L^1} = \mathrm{TV}(\tilde{\rho},\ \rho_\mu), \tag{4.3.5}$$

the fraction of misallocated mass. The same change of variables identifies this in the continuum with the marginal discrepancy $\mathrm{TV}(T_{\#}\mu, \nu)$ of (4.3.3), so Tables 5 and 7 estimate one quantity by two different discretizations. Table 7 reports the aggregated residuals across all distributions and grids.

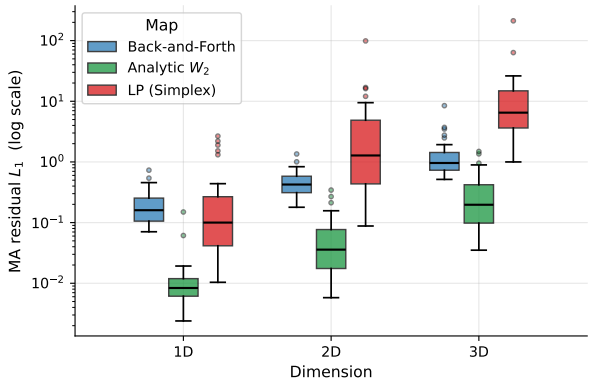


FIGURE 19. *Monge–Ampère residual (L^1 norm, log scale) of three maps on Gaussian instances $\mathcal{N}(m, \sigma_\nu^2 I_d)$.* Analytic *(green) is the exact map (Takatsu, 2011), whose residual is the pure discretization floor;* LP (simplex) *(red) is the barycentric projection of the exact plan;* Back-and-Forth *(blue) is the native map. For $d \geq 2$ Back-and-Forth lies below LP; all three sit above the floor in 3D.*

Two effects contribute to the measured residual, and separating them is essential for reading Table 7. The first is the error of the map. The second is a discretization artifact: on a finite grid the term $\rho_\nu(T(x))$ must be interpolated at off-grid points $T(x)$, which introduces an $O(h^2)$ error even for an exact map. We make this precise in Appendix A.4, where we derive the discretization floor $\|r\|_{L^1} \approx \frac{h^2}{12}\, \mathbb{E}\big[\, |\chi^2(d) - d| \,/\, \sigma_\nu^2 \,\big]$ for Gaussian targets $\mathcal{N}(m, \sigma_\nu^2 I_d)$ and confirm it numerically to within 5%. Since this floor grows with d, much of the residual increase from 1D to 3D in Table 7 is discretization, not solver degra-

TABLE 7

Monge–Ampère residuals (L^1 norm), pooled over grid sizes and distributions. Entries report the median residual and interquartile range (IQR) over converged runs. The final column gives the associated misallocated-mass fraction, $\frac{1}{2}\|r\|_{L^1}$. Bold marks the lowest median residual in each dimension. Adaptive with $\sigma_s = 1$ is lowest throughout.

Dim.	Method	σ_s	Median residual	IQR	Misalloc. mass
1D	CIC	1	0.076	[0.041, 0.151]	3.8%
	CIC	8	0.099	[0.049, 0.198]	5.0%
	Adaptive	1	**0.067**	[0.040, 0.129]	3.4%
	Adaptive	8	0.120	[0.066, 0.216]	6.0%
2D	CIC	1	0.196	[0.125, 0.319]	9.8%
	CIC	8	0.235	[0.147, 0.407]	11.8%
	Adaptive	1	**0.165**	[0.092, 0.287]	8.3%
	Adaptive	8	0.191	[0.107, 0.364]	9.6%
3D	CIC	1	0.442	[0.290, 0.585]	22.1%
	CIC	8	0.440	[0.315, 0.620]	22.0%
	Adaptive	1	**0.349**	[0.207, 0.530]	17.5%
	Adaptive	8	0.415	[0.236, 0.700]	20.8%

dation; the residuals are therefore read relative to the floor and used to rank configurations rather than as absolute errors. On Gaussian instances, where the optimal map is known analytically, we separate the two effects directly (Figure 19). The analytic map's residual is the pure discretization floor; that the floor rises with dimension alongside the solver residuals confirms the increase is discretization, not solver error. The barycentric averaging of the split LP targets (Remark A.2) compresses that map and inflates its residual, which is why Back-and-Forth lies below LP for $d \geq 2$.

Taken together, the Back-and-Forth method occupies a distinct niche in the benchmark. It is the only solver that returns a transport map directly. Its structural output is sound: the recovered map is monotone almost everywhere, and on Gaussian instances its Monge–Ampère residuals lie below those of the barycentric LP projection. The cost underestimation and residual growth with dimension are discretization artifacts, not algorithmic failures, and shrink under grid refinement. Throughout, the pushforward procedure is the cornerstone of the method.

4.4 Application: 2−Wasserstein barycenter for continuous measures

Let $\Omega \subset \mathbb{R}^d$ be a compact subset of $\mathbb{R}^d$, $\mu_1, \dots, \mu_N \in \mathcal{P}(\Omega)$ be probability measures and $\lambda_1, \dots, \lambda_N \geq 0$ be non-negative numbers with $\sum_{i=1}^N \lambda_i = 1$.

Definition 4.4.1. The 2−Wasserstein Barycenter of $(\mu_i)_{i=1}^N$ with weights $(\lambda_i)_{i=1}^N$ is defined as a minimizer (Agueh and Carlier, 2011)

$$\nu^\star \in \arg\min_{\nu \in \mathcal{P}(\Omega)} \sum_{i=1}^N \lambda_i \, W_2^2(\mu_i, \nu). \tag{4.4.1}$$

Outer regularization: In practice, it is common to (outer-)regularize the 2−Wasserstein Barycenter (4.4.1) (see, e.g., Li et al., 2020)

$$\nu_\gamma^\star \in \arg\min_{\nu \in \mathcal{P}(\Omega)} \mathrm{Bar}_\gamma(\nu; \mu_1, \dots, \mu_N) := \arg\min_{\nu \in \mathcal{P}(\Omega)} \sum_{i=1}^N \lambda_i \, W_2^2(\mu_i, \nu) \; + \; \gamma \mathcal{R}(\nu), \tag{4.4.2}$$

where

$$\mathcal{R}(\nu) := \begin{cases} \int_\Omega \nu(y)\big(\ln \nu(y) - 1\big)\, \mathrm{d}y & \nu \text{ is absolutely continuous w.r.t. Lebesgue,} \\ +\infty & \text{otherwise.} \end{cases} \tag{4.4.3}$$

The outer-regularization enforces absolute continuity of the minimizer $\nu_\gamma^\star$ in (4.4.2), which can, for instance, improve convergence of our computational algorithms. Indeed, notice that $\nu \mapsto \mathrm{Bar}_\gamma(\nu; \cdot)$ is strictly convex and the first variation is given by Chizat (2025)

$$\nu_\gamma^\star \propto \exp\Big(-\tfrac{1}{\gamma} \sum_{i=1}^N \lambda_i \, \psi_i(y)\Big),$$

where ψ_i are the Kantorovich potentials associated to $W_2(\mu_i, \nu)$.

Dual formulation: The following theorem provides a dual formulation for the outer-regularized 2−Wasserstein Barycenter in (4.4.2).

Theorem 4.4.2 (Duality)**.** *Let $\Omega \subset \mathbb{R}^d$ be a compact and convex subset, $\mu_1, \dots, \mu_N \in \mathcal{P}(\Omega)$ be probability measures, $\lambda_1, \dots, \lambda_N \geq 0$ be non-negative numbers with $\sum_{i=1}^N \lambda_i = 1$, and $\gamma > 0$ be a positive number. Then the outer-regularized 2− Wasserstein Barycenter admits the dual formulation*

$$\min_{\nu \in \mathcal{P}(\Omega)} \mathrm{Bar}_\gamma(\nu; \mu_1, \dots, \mu_N) = \sup_{\psi_1, \dots, \psi_N} \left\{ \sum_{i=1}^N \lambda_i \int_\Omega \psi_i^c(x)\, d\mu_i(x) \right. \\ \left. -\gamma \ln\left(\int_\Omega \exp\left(-\frac{1}{\gamma} \sum_{i=1}^N \lambda_i \psi_i(y) \right) \mathrm{d}y \right) \right\}.$$

Proof. By Kantorovich duality (Theorem 2.2.3; see also Santambrogio (2015)), each squared distance admits the dual representation

$$W_2^2(\mu_i, \nu) = \sup_{\psi_i} \left\{ \int_\Omega \psi_i^c(x)\, d\mu_i(x) + \int_\Omega \psi_i(y)\, \mathrm{d}\nu(y) \right\}. \tag{4.4.4}$$

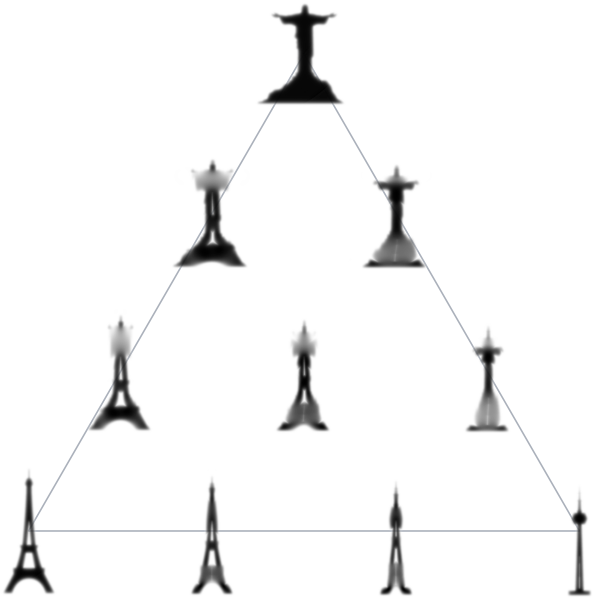


FIGURE 20. *Wasserstein barycenters of three landmark silhouettes — Christ the Redeemer, the Eiffel Tower, and CN Tower — placed at the vertices of the weight simplex. Each interior image is the barycenter whose barycentric weights are given by its position in the triangle, computed with the barycenter-regularized Back-and-Forth method on a* 1024×1024 *grid with* $\gamma = 10^{-4}$.

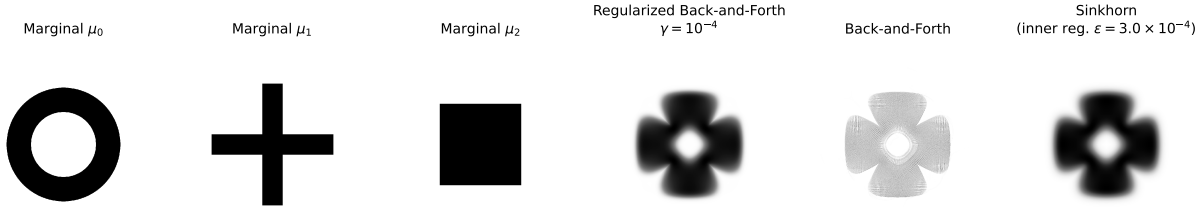


FIGURE 21. *Equal-weight barycenter* ($\lambda_0 = \lambda_1 = \lambda_2 = 1/3$) *of three shapes: a ring, a cross, and a square. From left to right: the input marginals* μ_0, μ_1, μ_2*, followed by the barycenter computed by the regularized back-and-forth method (§B.5) at outer regularization* $\gamma = 10^{-4}$*, the plain back-and-forth method, and convolutional Sinkhorn (§B.4) with inner regularization* $\varepsilon = 3.0 \times 10^{-4}$*. The plain back-and-forth output is noisy, whereas both regularized methods produce smooth densities.*

Substituting (4.4.4) into (4.4.2) gives

$$\begin{aligned}
\min_{\nu \in \mathcal{P}(\Omega)} \mathrm{Bar}_\gamma(\nu; \mu_1, \dots, \mu_N) &= \inf_{\nu \in \mathcal{P}(\Omega)} \sup_{\psi_i \in L^1(\Omega)} \sum_{i=1}^N \lambda_i \left(\int_\Omega \psi_i^c(x)\, d\mu_i + \int_\Omega \psi_i(y)\, d\nu \right) \\
&\qquad + \gamma \int \nu(y)(\ln \nu(y) - 1)\, \mathrm{d}y \\
&= \sup_{\psi_i \in L^1(\Omega)} \inf_{\nu \in \mathcal{P}(\Omega)} \sum_{i=1}^N \lambda_i \left(\int_\Omega \psi_i^c(x)\, d\mu_i + \int_\Omega \psi_i(y)\, d\nu \right) \\
&\qquad + \gamma \int \nu(y)(\ln \nu(y) - 1)\, \mathrm{d}y
\end{aligned}$$

$$= \sup_{\psi_i \in L^1(\Omega)} \int_\Omega \sum_{i=1}^N \lambda_i \psi_i^c(x)\, d\mu_i + \inf_{\nu \geq 0,\ \int \nu = 1} \int_\Omega \Big(\sum_{i=1}^N \lambda_i \psi_i(y) \Big) \nu(y)\, \mathrm{d}y + \gamma \int_\Omega \nu(y) \big(\ln \nu(y) - 1 \big)\, \mathrm{d}y.$$

where we have used Sion's Theorem A.1, to exchange $\min_\nu$ with $\sup_{\psi_1,\dots,\psi_N}$ [5].
Now, we claim that

$$\inf_{\nu \geq 0,\ \int \nu = 1} \int_\Omega \Big(\sum_{i=1}^N \lambda_i \psi_i(y) \Big) \nu(y)\, \mathrm{d}y + \gamma \int_\Omega \nu(y) \big(\ln \nu(y) - 1 \big)\, \mathrm{d}y = -\gamma \ln \left(\int_\Omega \exp\left(-\frac{1}{\gamma} \sum_{i=1}^N \lambda_i \psi_i(y) \right) \mathrm{d}y \right) - \gamma. \quad (4.4.5)$$

Indeed, taking the first variation of (4.4.5), one has

$$\sum_{i=1}^N \lambda_i \psi_i(y) + \gamma \ln \nu(y) = c, \qquad y \in \Omega, \quad (4.4.6)$$

for some constant $c \in \mathbb{R}$. Hence,

$$\nu(y) = \exp\left(\frac{c - \sum_{i=1}^N \lambda_i \psi_i(y)}{\gamma} \right) = C \exp\left(-\frac{1}{\gamma} \sum_{i=1}^N \lambda_i \psi_i(y) \right), \quad (4.4.7)$$

with normalization

$$C^{-1} = \int_\Omega \exp\left(-\frac{1}{\gamma} \sum_{i=1}^N \lambda_i \psi_i(y) \right) \mathrm{d}y. \quad (4.4.8)$$

Plugging (4.4.7) back into (4.4.5) shows that minimizing over ν contributes the log-partition term

$$-\gamma \ln \int \exp\big(-\tfrac{1}{\gamma} \sum_i \lambda_i \psi_i \big),$$

together with the constant $-\gamma$. The latter is independent of the potentials ψ_i (and of ν): it shifts the objective by a fixed amount without affecting either the minimizer $\nu_\gamma^\star$ or the supremum below, so we drop it. (Equivalently, the constant reflects the normalization -1 in (4.4.3); with the choice $\mathcal{R}(\nu) = \int_\Omega \nu \ln \nu$, which differs from (4.4.3) by a constant on $\mathcal{P}(\Omega)$, the identity (4.4.5) holds without

[5] The exchange of $\inf_\nu$ and $\sup_{\psi_1,\dots,\psi_N}$ is carried out at the formal level: we invoke Sion's minimax theorem (Theorem A.1) without verifying all of its hypotheses—joint convexity in ν and concavity in the ψ_i, compactness of the ν-domain, and the necessary semicontinuity. A fully rigorous treatment would establish these conditions but we do not pursue it here.

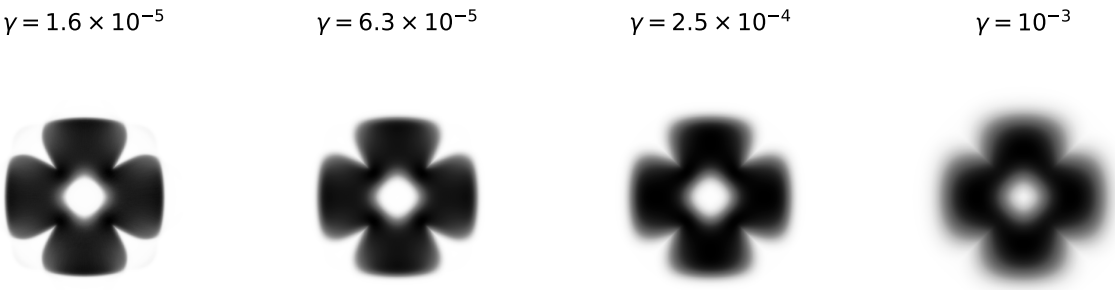


FIGURE 22. *Effect of the outer regularization γ on the Back-and-Forth barycenter. As γ grows from 1.6×10^{-5} to 10^{-3}, the grid-scale noise of the weakly regularized map is progressively suppressed and the barycenter becomes smoother, at the cost of blurring the sharp features of the shape.*

the trailing $-\gamma$.) Consequently,

$$\min_{\nu \in \mathcal{P}(\Omega)} \mathrm{Bar}_\gamma(\nu; \mu_1, \dots, \mu_N) = \sup_{\psi_1, \dots, \psi_N} \left\{ \sum_{i=1}^N \lambda_i \int_\Omega \psi_i^c(x)\, d\mu_i(x) \right. \\ \left. -\gamma \ln\left(\int_\Omega \exp\left(-\frac{1}{\gamma} \sum_{i=1}^N \lambda_i \psi_i(y) \right) \mathrm{d}y \right) \right\}. \quad (4.4.9)$$

□

Remark 4.4.3. Equivalently, this term couples all potentials $\{\psi_j\}$ through a log-partition function, and can be viewed as a smooth "soft-min" version of the hard constraint/aggregation that would appear in the unregularized case $\gamma \to 0$.

4.4.1 (Outer-)regularized Back-and-Forth for $2-$Wasserstein barycenter

We will use a single convention: φ-potentials live on the source (so $\varphi = \varphi(x)$) and ψ-potentials live on the target (so $\psi = \psi(y)$). Let $\psi_i : \Omega \to \mathbb{R}$ and collect them into the vector $\boldsymbol{\psi} := (\psi_1, \dots, \psi_N)$. Define the corresponding c-transforms

$$\varphi_i(x) := \psi_i^c(x) := \inf_{y \in \Omega} \big\{ c(x, y) - \psi_i(y) \big\}, \qquad i = 1, \dots, N, \quad (4.4.10)$$

and write $\boldsymbol{\varphi} := (\varphi_1, \dots, \varphi_N)$. In the context of Euclidean cost $c(x, y) = \frac{1}{2}\|x - y\|^2$, by Brenier's theorem the associated Monge map (from μ_i to ν) is $T_i(x) = x - \nabla\varphi_i(x)$. Thus it is natural that we define that φ_i and ψ_i live in the Sobolev space

$$H^1(\Omega) = \big\{ u \in L^2(\Omega) : \nabla u \in L^2(\Omega) \big\}, \qquad \langle u, v \rangle_{H^1} = \int_\Omega \big(uv + \nabla u \cdot \nabla v \big)\, \mathrm{d}x. \quad (4.4.11)$$

In terms of $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$, we define the dual functionals $F : (H^1)^N \to \mathbb{R}$ and $D : (H^1)^N \to \mathbb{R}$

$$\begin{aligned} F(\boldsymbol{\psi}) &:= \sum_{i=1}^N \lambda_i \int_\Omega \psi_i^c(x)\, d\mu_i(x) - \gamma \ln\left(\int_\Omega \exp\left(-\frac{1}{\gamma}\sum_{i=1}^N \lambda_i \psi_i(y)\right) \mathrm{d}y\right), \\ D(\boldsymbol{\varphi}) &:= \sum_{i=1}^N \lambda_i \int_\Omega \varphi_i(x)\, d\mu_i(x) - \gamma \ln\left(\int_\Omega \exp\left(-\frac{1}{\gamma}\sum_{i=1}^N \lambda_i \varphi_i^c(y)\right) \mathrm{d}y\right). \end{aligned} \tag{4.4.12}$$

Whenever $\varphi_i = \psi_i^c$ (as in (4.4.10)), the two coincide.

Proposition 4.4.4. *For every $\boldsymbol{h} = (h_1, \dots, h_N) \in (H^1(\Omega))^N$:*

(i) The first variations of the functionals F and D are given by

$$\delta F(\boldsymbol{\psi})[\boldsymbol{h}] = \sum_{i=1}^N \lambda_i \int_\Omega h_i(y)\, \mathrm{d}\nu - \sum_{i=1}^N \lambda_i \int_\Omega h_i(y)\, \mathrm{d}(T_i)_{\#}\mu_i \tag{4.4.13}$$

$$\delta D(\boldsymbol{\varphi})[\boldsymbol{h}] = \sum_{i=1}^N \lambda_i \int_\Omega h_i(x)\, d\mu_i(x) - \sum_{i=1}^N \lambda_i \int_\Omega h_i(x_i^*(y))\, d\nu(y), \tag{4.4.14}$$

where the measure ν is defined as

$$\mathrm{d}\nu := \frac{\exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j \varphi_j^c(y)\Big)\, \mathrm{d}y}{\int_\Omega \exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j \varphi_j^c(y)\Big)\, \mathrm{d}y}.$$

(ii) Their Sobolev gradients can be formally written as

$$\begin{aligned} \nabla_{H^1} F(\boldsymbol{\psi})_i &= (I - \Delta)^{-1}\Big(\lambda_i\big(\nu - (T_i)_{\#}\mu_i\big)\Big), \\ \nabla_{H^1} D(\boldsymbol{\varphi})_i &= (I - \Delta)^{-1}\Big(\lambda_i\big(\mu_i - (S_i)_{\#}\nu\big)\Big). \end{aligned}$$

Proof. **Finding a Sobolev-space derivative.** Let $\boldsymbol{h} = (h_1, \dots, h_N) \in (H^1(\Omega))^N$ and consider the perturbation $\boldsymbol{\varphi}_\varepsilon := \boldsymbol{\varphi} + \varepsilon \boldsymbol{h}$. Our goal is to find

$$\delta D(\boldsymbol{\varphi})[\boldsymbol{h}] = \frac{d}{d\varepsilon} D(\boldsymbol{\varphi} + \varepsilon \boldsymbol{h})\Big|_{\varepsilon=0},$$

where

$$D(\boldsymbol{\varphi}) = \sum_{i=1}^N \lambda_i \int_\Omega \varphi_i(x)\, d\mu_i(x) - \gamma \ln Z(\boldsymbol{\varphi}), \qquad Z(\boldsymbol{\varphi}) := \int_\Omega \exp\left(-\frac{1}{\gamma}\sum_{i=1}^N \lambda_i \varphi_i^c(y)\right) \mathrm{d}y. \tag{4.4.15}$$

Derivative of the linear term. By linearity of the integral,

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} \sum_{i=1}^N \lambda_i \int_\Omega \big(\varphi_i(x) + \varepsilon h_i(x)\big)\, d\mu_i(x) = \sum_{i=1}^N \lambda_i \int_\Omega h_i(x)\, d\mu_i(x). \tag{4.4.16}$$

Derivative of the log-partition term (chain rule). We first apply the chain rule to $-\gamma \ln Z(\boldsymbol{\varphi}_\varepsilon)$:

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} \big[-\gamma \ln Z(\boldsymbol{\varphi}_\varepsilon)\big] = -\gamma \, \frac{1}{Z(\boldsymbol{\varphi})} \; \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} Z(\boldsymbol{\varphi}_\varepsilon). \tag{4.4.17}$$

Next, we differentiate Z under the integral sign:

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} Z(\boldsymbol{\varphi}_\varepsilon) = \int_\Omega \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} \exp\left(-\frac{1}{\gamma}\sum_{i=1}^N \lambda_i \big(\varphi_i + \varepsilon h_i\big)^c(y)\right) \mathrm{d}y \tag{4.4.18}$$

$$= \int_\Omega \exp\left(-\frac{1}{\gamma}\sum_{i=1}^N \lambda_i \varphi_i^c(y)\right) \left(-\frac{1}{\gamma}\sum_{i=1}^N \lambda_i \; \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} \big(\varphi_i + \varepsilon h_i\big)^c(y)\right) \mathrm{d}y. \tag{4.4.19}$$

Envelope step for the c-transform. For each i and $y \in \Omega$, let

$$x_i^\star(y) \in \arg\min_{x\in\Omega} \big\{c(x,y) - \varphi_i(x)\big\}, \tag{4.4.20}$$

assuming the minimizer $x_i^\star(y)$ exists and is unique. Then we can write the envelope step in three stages:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} \big(\varphi_i + \varepsilon h_i\big)^c(y) &= \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} \min_{x\in\Omega} \Big\{c(x,y) - \varphi_i(x) - \varepsilon h_i(x)\Big\} \\ &= \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} \Big(c\big(x_i^\star(y), y\big) - \varphi_i\big(x_i^\star(y)\big) - \varepsilon h_i\big(x_i^\star(y)\big)\Big) \\ &= -h_i\big(x_i^\star(y)\big). \end{aligned} \tag{4.4.21}$$

Substituting (4.4.21) into (4.4.18) and then into (4.4.17) yields

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} \big[-\gamma \ln Z(\boldsymbol{\varphi}_\varepsilon)\big] = -\sum_{i=1}^N \lambda_i \, \frac{\int_\Omega h_i\big(x_i^\star(y)\big) \, \exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j \varphi_j^c(y)\Big) \, \mathrm{d}y}{\int_\Omega \exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j \varphi_j^c(y)\Big) \, \mathrm{d}y}. \tag{4.4.22}$$

First variation. Combining the derivative of the linear term with (4.4.22) we obtain the Gâteaux derivative (in the sense of Definition 3.4.2)

$$\begin{aligned} \delta D(\boldsymbol{\varphi})[\boldsymbol{h}] &= \sum_{i=1}^N \lambda_i \int_\Omega h_i(x) \, d\mu_i(x) - \sum_{i=1}^N \lambda_i \, \frac{\int_\Omega h_i\big(x_i^\star(y)\big) \, \exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j \varphi_j^c(y)\Big) \, \mathrm{d}y}{\int_\Omega \exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j \varphi_j^c(y)\Big) \, \mathrm{d}y}. \\ &= \sum_{i=1}^N \lambda_i \int_\Omega h_i(x) \, d\mu_i(x) - \sum_{i=1}^N \lambda_i \int_\Omega h_i(x_i^*(y)) \, d\nu(y) \end{aligned} \tag{4.4.23}$$

where the measure ν is defined as

$$\mathrm{d}\nu := \frac{\exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j \varphi_j^c(y)\Big) \, \mathrm{d}y}{\int_\Omega \exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j \varphi_j^c(y)\Big) \, \mathrm{d}y}.$$

Now, note that $x_i^\star(y) \in \arg\min_{x\in\Omega}\{c(x,y)-\varphi_i(x)\}$ defines, for each i, a (measurable) map $S_i : \Omega \to \Omega$ by $S_i(y) := x_i^\star(y)$. When the potentials are linked by the c-transform (i.e. when $\varphi_i = \psi_i^c$ so that $D(\boldsymbol{\varphi}) = F(\boldsymbol{\psi})$ in value), the corresponding optimal Monge maps $T_i : \Omega \to \Omega$ satisfy $T_i \circ S_i = \mathrm{Id}$ and hence $S_i = T_i^{-1}$ (a.e.). Thus, in this case we can use the change-of-variables formula to substitute $y = T_i(x)$ inside the integral:

$$\delta D(\boldsymbol{\varphi})[\boldsymbol{h}] = \sum_{i=1}^N \lambda_i \int_\Omega h_i(x)\, d\mu_i(x) - \sum_{i=1}^N \lambda_i \int_\Omega h_i(x)\, d\nu(T_i(x)). \tag{4.4.24}$$

By the definition of pushforward, $\int_\Omega h_i(x)\,\mathrm{d}\nu(T_i(x)) = \int_\Omega h_i\,\mathrm{d}\big((S_i)_\#\nu\big)$, so that

$$\delta D(\boldsymbol{\varphi})[\boldsymbol{h}] = \sum_{i=1}^N \lambda_i \int_\Omega h_i(x)\, d\mu_i(x) - \sum_{i=1}^N \lambda_i \int_\Omega h_i(x)\,\mathrm{d}\big((S_i)_\#\nu\big). \tag{4.4.25}$$

In the same c-transform setting described above, rewriting (4.4.25) in target variables yields the equivalent first-variation statement used below:

$$\delta F(\boldsymbol{\varphi})[\boldsymbol{h}] = \sum_{i=1}^N \lambda_i \int_\Omega h_i(y)\,\mathrm{d}\nu - \sum_{i=1}^N \lambda_i \int_\Omega h_i(y)\,\mathrm{d}(T_i)_\#\mu_i \tag{4.4.26}$$

Hence the Sobolev-space gradients (with respect to $\langle\cdot,\cdot\rangle_{H^1}$) can be written formally as

$$\nabla_{H^1} F(\boldsymbol{\psi})_i = (I-\Delta)^{-1}\Big(\lambda_i\big(\nu - (T_i)_\#\mu_i\big)\Big),$$
$$\nabla_{H^1} D(\boldsymbol{\varphi})_i = (I-\Delta)^{-1}\Big(\lambda_i\big(\mu_i - (S_i)_\#\nu\big)\Big).$$

□

4.4.2 Computational algorithm: (outer-)regularized Back-and-Forth

The Proposition 4.4.4 suggests a novel computational algorithm computing numerical realizations of (outer-regularized) 2−Wasserstein Barycenters via alternate maximization of the functionals F and D.

Following the approach of the Back-and-Forth method discussed in Section 3.4, let $\{\psi_i^{(k)}\}_{k\geq 0}$ and $\{\varphi_i^{(k)}\}_{k\geq 0}$ be a (formal) sequence of potentials obtained via Sobolev gradient-ascent iterations of the functionals F and D as follows

$$\psi_i^{(k+1/2)} \longleftarrow \psi_i^{(k)} + \eta_k\, \nabla_{H^1} F\big(\boldsymbol{\psi}^{(k)}\big)_i = \tag{4.4.27}$$
$$\psi_i^{(k)} + \eta_k\,(I-\Delta)^{-1}\Big(\lambda_i\big(\nu^{(k)} - (T_i^{(k)})_\#\mu_i\big)\Big), \tag{4.4.28}$$
$$\varphi_i^{(k+1/2)} \longleftarrow \big(\psi_i^{(k+1/2)}\big)^c, \tag{4.4.29}$$
$$\varphi_i^{(k+1)} \longleftarrow \varphi_i^{(k+1/2)} + \eta_k\, \nabla_{H^1} D\big(\boldsymbol{\varphi}^{(k+1/2)}\big)_i = \tag{4.4.30}$$
$$\varphi_i^{(k+1/2)} + \eta_k\,(I-\Delta)^{-1}\Big(\lambda_i\big(\mu_i - (S_i^{(k+1/2)})_\#\nu^{(k+1/2)}\big)\Big), \tag{4.4.31}$$
$$\psi_i^{(k+1)} \longleftarrow \big(\varphi_i^{(k+1)}\big)^c, \tag{4.4.32}$$

for $i = 1, \dots, N$.
Define the associated barycenter iterate $\nu^{(k)}$ by the Gibbs rule

$$\nu^{(k)}(\mathrm{d}y) := \frac{\exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j \psi_j^{(k)}(y)\Big)}{\int_\Omega \exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j \psi_j^{(k)}(y)\Big)\,\mathrm{d}y}\,\mathrm{d}y.$$

and consider $T_i^{(k)}$ the Monge maps from μ_i to $\nu^{(k)}$. Then, with a step size $\eta_k > 0$, we update

$$\psi_i^{(k+1)} = \psi_i^{(k)} + \eta_k\,(I-\Delta)^{-1}\Big(\lambda_i\big(\nu^{(k)} - (T_i^{(k)})_\# \mu_i\big)\Big), \qquad i = 1, \dots, N,$$
$$\varphi_i^{(k+1)} = \varphi_i^{(k)} + \eta_k\,(I-\Delta)^{-1}\Big(\lambda_i\big(\mu_i - (S_i^{(k)})_\# \nu^{(k)}\big)\Big), \qquad i = 1, \dots, N,$$

where $S_i^{(k)}(y) := (x_i^\star)^{(k)}(y)$ denotes the corresponding minimizer map in the c-transform.

The pseudo-code and a detailed version of the (outer-)regularized Back-and-Forth algorithms is provided below 3.

Algorithm 3 (Outer-)Regularized Back-and-Forth Dual Ascent (Sobolev Gradient)

Input: measures $\mu_1, \dots, \mu_N$, weights $\lambda_1, \dots, \lambda_N$, regularization $\gamma > 0$, step sizes $\{\eta_k\}_{k \geq 0}$
Initialize: potentials $\boldsymbol{\psi}^{(0)} = \mathbf{0}$ and $\boldsymbol{\varphi}^{(0)} = \mathbf{0}$ (zero tensors/functions)
for $k = 0, 1, 2, \dots$ **do** ▷ $F(\boldsymbol{\psi})$-view (target potentials)
  for $i = 1, \dots, N$ **do**
    $\varphi_i^{(k)} \leftarrow (\psi_i^{(k)})^c$
  end for
  $\nu^{(k)}(\mathrm{d}y) \leftarrow \dfrac{\exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j\,\psi_j^{(k)}(y)\Big)}{\int_\Omega \exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j\,\psi_j^{(k)}(y)\Big)\,\mathrm{d}y}\,\mathrm{d}y$
  for $i = 1, \dots, N$ **do**
    compute $T_i^{(k)}$ from the source potential $\varphi_i^{(k)}$ ▷ e.g. $T_i^{(k)}(x) = x - (\nabla w)^{-1}(\nabla \varphi_i^{(k)}(x))$
    $\psi_i^{(k+1)} \leftarrow \psi_i^{(k)} + \eta_k\,(I-\Delta)^{-1}\Big(\lambda_i\big(\nu^{(k)} - (T_i^{(k)})_\# \mu_i\big)\Big)$
  end for
  ▷ $D(\boldsymbol{\varphi})$-view (source potentials)
  for $i = 1, \dots, N$ **do**
    $\psi_i^{(k)} \leftarrow (\varphi_i^{(k)})^c$
  end for
  $\nu^{(k)}(\mathrm{d}y) \leftarrow \dfrac{\exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j\,(\varphi_j^{(k)})^c(y)\Big)}{\int_\Omega \exp\Big(-\frac{1}{\gamma}\sum_{j=1}^N \lambda_j\,(\varphi_j^{(k)})^c(y)\Big)\,\mathrm{d}y}\,\mathrm{d}y$
  for $i = 1, \dots, N$ **do**
    compute $S_i^{(k)}$ from the target potential $(\varphi_i^{(k)})^c$ ▷ e.g. $S_i^{(k)}(y) = y - (\nabla w)^{-1}(\nabla (\varphi_i^{(k)})^c(y))$
    $\varphi_i^{(k+1)} \leftarrow \varphi_i^{(k)} + \eta_k\,(I-\Delta)^{-1}\Big(\lambda_i\big(\mu_i - (S_i^{(k)})_\# \nu^{(k)}\big)\Big)$
  end for
end for

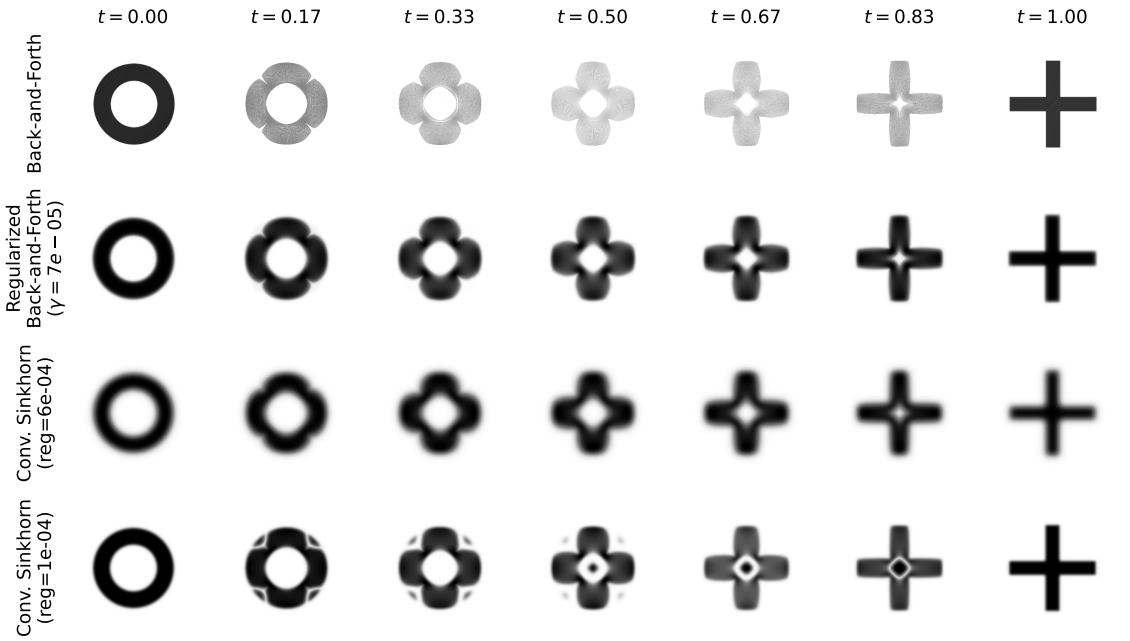


FIGURE 23. *Barycentric interpolation between two shapes (ring and cross) as the weight $\lambda_0 = t$ sweeps from 1 to 0 (with $\lambda_1 = t - 1$). Rows: the (outer-)regularized Back-and-Forth method, the plain Back-and-Forth method (see Appendix B.5), and the convolutional Sinkhorn (see Appendix B.4) at two inner-regularization levels ($\varepsilon = 6 \times 10^{-4}$ and $\varepsilon = 10^{-4}$). The plain Back-and-Forth output is noisy; convolutional Sinkhorn is overly blurred at the larger ε and develops artifacts at the smaller one. The outer-regularized variant interpolates cleanly without either defect.*

Convergence criterion (marginal mismatch). To monitor convergence in practice (and, indirectly, the quality of the computed Monge maps S_i), we track a *marginal mismatch* error defined from the discrepancy between each reference marginal μ_i and the pushforward of the current barycenter iterate $\nu^{(k)}$ under $S_i^{(k)}$. Given an iterate $\nu^{(k)}$ and maps $S_i^{(k)}$, define

$$\mathsf{Err}_p^{(k)} := \max_{1 \le i \le N} \big\| \mu_i - (S_i^{(k)})_{\#} \nu^{(k)} \big\|_p, \qquad p \in \{1, 2\}. \tag{4.4.33}$$

Here $\|\cdot\|_p$ denotes a chosen L^p-type norm of the difference of densities (when available) or, more generally, a discretized norm on the numerical grid. We stop the iteration once $\mathsf{Err}_p^{(k)}$ falls below a prescribed tolerance (we usually set tolerance $\tau = 10^{-3}$, with $p = 1$).

4.4.3 A general framework

The (outer-)regularized barycenter is the simplest case of a broader family of variational problems on a graph of measures. Let $G = (V, E)$ be a directed graph with vertex set $V = \{0, 1, \dots, N\}$ and edge set $E \subset V \times V$; associate to each node $i \in V$ a measure $\rho_i \in \mathcal{M}_+(\Omega)$ and to each edge $(i, j) \in E$ a cost $c_{ij} \colon \Omega \times \Omega \to \mathbb{R}$. With a proper, convex, lower semicontinuous functional $\Phi \colon (\mathcal{M}_+(\Omega))^{N+1} \to \mathbb{R} \cup \{+\infty\}$ acting on the nodes, the problem reads

$$\inf_{\substack{\rho_i \in \mathcal{M}_+(\Omega),\, i \in V \\ \pi_{ij} \in \Pi(\rho_i, \rho_j),\, (i,j) \in E}} \sum_{(i,j) \in E} \int_{\Omega \times \Omega} c_{ij}(x_i, x_j)\, d\pi_{ij}(x_i, x_j) \;+\; \Phi\big((\rho_i)_{i \in V}\big), \tag{4.4.34}$$

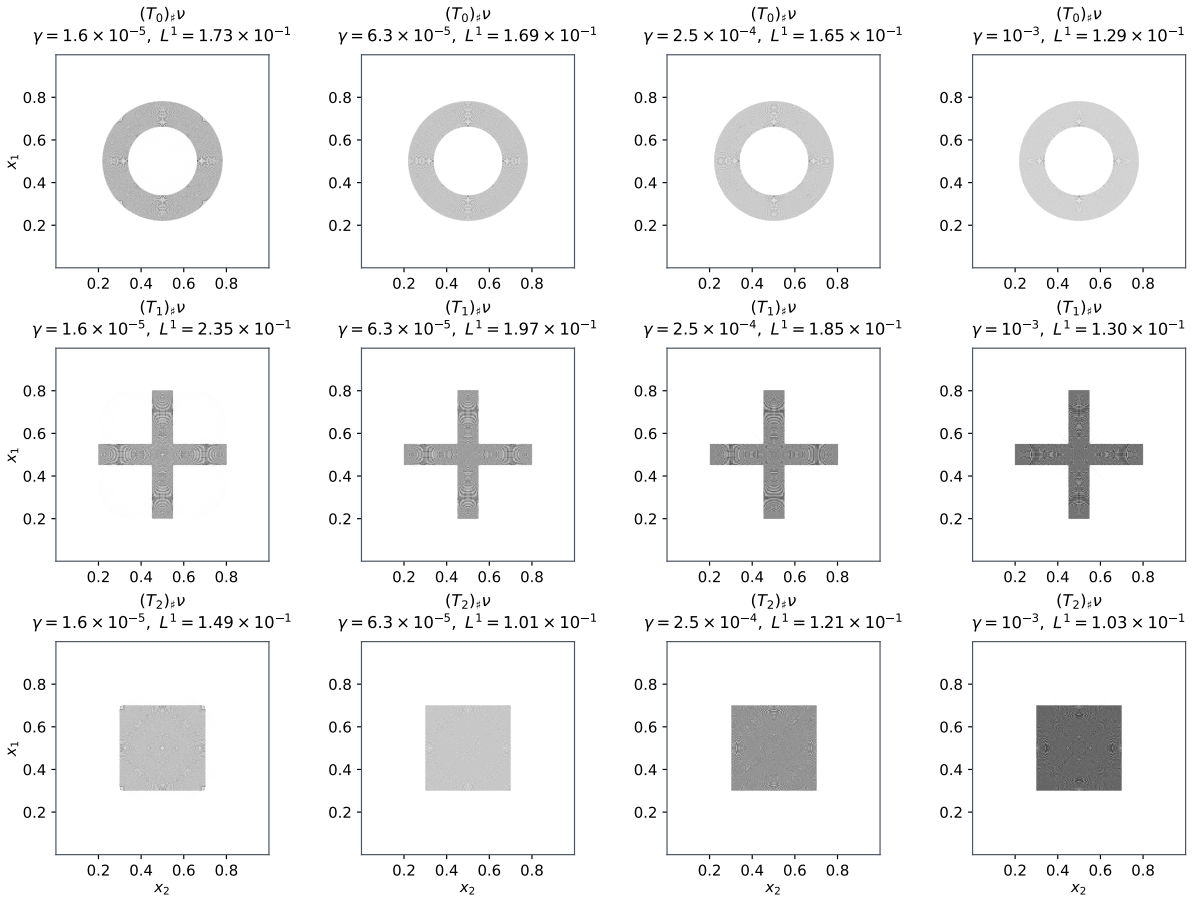


FIGURE 24. *Pushforward of the barycenter measure ν onto each input marginal through the recovered Monge maps T_0, T_1, T_2. Rows correspond to the three marginals (ring, cross, square), columns to increasing outer regularization γ. Each title reports the L^1 error between the pushforward $(T_k)_\#\nu$ and the target marginal. Pushforward fidelity improves markedly as γ grows, since a stronger outer regularization yields smoother, better-behaved maps.*

where each $\pi_{ij} \in \Pi(\rho_i, \rho_j)$ is a transport plan with marginals ρ_i and ρ_j. The barycenter problem (4.4.2) is the special case of a star graph: the centre node $\rho_0 = \nu$ is free, the leaves $\rho_i = \mu_i$ are fixed, the edge costs carry the weighted squared distances, and Φ collects the outer entropy on the centre together with the constraints fixing the leaves. We develop a Back-and-Forth Flow for the general problem (4.4.34) in joint work with D. Ruban and A. Gerolin (Zhytkevych et al., 2026): the same dual ascent applies, now with one potential per edge. This manuscript is currently under review, and the duality and variational derivations there are carried out at a formal level.

4.5 Application: colour transfer

In colour transfer applications, the goal is to recolour an image to match a desired colour palette. This technique finds use in fields such as computer graphics, video postprocessing, panorama stitching, medical imaging, data visualization, and more (Faridul et al., 2016; Pitié, 2020). OT is one of the statistical approaches to colour transfer. Unlike per-channel methods — which can introduce unnatural visual artifacts — the OT approach aligns the colour histograms jointly in a three-dimensional space, preserving cross-channel correlations (Frigo et al., 2015). In statistical colour transfer, the colour distributions of both images are treated as probability distributions, and the transformation is performed by matching these histograms. Here, the "source image" refers to the image being recoloured, while the "target image" supplies the colour palette to be transferred. We denote by μ the measure associated with the source image and by ν the measure associated with the target image.

To evaluate the performance of optimal transport methods for color transfer, we utilize the MIT-Adobe FiveK dataset (Bychkovsky et al., 2011) for benchmarking experiments.[6]

Setting. Let the two empirical colour distributions of the source and target images, μ and ν, be represented as histograms living on the colour space $\Omega \subset \mathbb{R}^d$. Concretely, $\mu = \sum_{i=1}^{n} a_i \, \delta_{x_i}$ and $\nu = \sum_{j=1}^{m} b_j \, \delta_{y_j}$, where $\{x_i\}$ and $\{y_j\}$ are the colour-bin centres (the support points) and $a_i, b_j \geq 0$ are the corresponding normalized bin frequencies, with $\sum_i a_i = \sum_j b_j = 1$. The support Ω is the RGB or CIELAB colour domain, with up to 3 channels.

A colour space is said to be perceptually-uniform if the equal differences in that space correspond to the equal perceptual differences as perceived by the human vision. This property is particularly important in tasks of image correction or colour transfer tasks, because the goal is not only to match the distributions numerically but also to preserve or reproduce the differences meaningfully.

CIELAB decomposes into a lightness component L and two chromatic components A (green-red) and B (blue-yellow), with ranges $[0, 100]$ for L and $[-128, 127]$ for A and B. This suggests three options: $d = 1$ (the L channel, adjusting only lightness), $d = 2$ (the A, B channels, transferring colour without touching lightness), or $d = 3$ (all channels). We use the 2-channel adjustment of A and B. CIELAB colour space was designed achieve the perceptual uniformity: the Euclidean differences between two colours (known as ΔE metric) in this space provides approximation of human-perceived difference. The difference ΔE between two colours (L_1, A_1, B_1) and (L_2, A_2, B_2) in CIELAB space is defined as

$$\Delta E = \sqrt{(L_2 - L_1)^2 + (A_2 - A_1)^2 + (B_2 - B_1)^2},$$

for which $\Delta E \approx 2.3$ corresponds to just noticeable difference. As a result, the Euclidean distance in CIELAB aligns the optimal transport geometry with perceptual similarity.

RGB has 3 channels — red, green and blue, taking integer values from 0 to 255 — so its measure representation uses all three ($d = 3$). RGB is not perceptually uniform, as Euclidean differences in that space do not correspond linearly to perceived colour differences. However, RGB remains the commongly used representation due to its simplicity, and including RGB-based experiments allow us to catch the robustness of optimal transport solvers under non-perceptual geometry.

Because the components of RGB and CIELAB have different bounds, we normalize each into the support $[0, 1]^d$. For both colour spaces we employ the squared Euclidean distance $c(x, y) = \|x - y\|^2$

[6]The photographs are copyright © 2011 Adobe Systems Incorporated and Massachusetts Institute of Technology, and are reproduced here under the dataset's Research License, which permits their use for non-commercial research purposes.

as the cost.

Brenier's map estimation: For the OT methods that output a transport plan $\pi \in \mathcal{P}(\Omega \times \Omega)$ (such as simplex and Sinkhorn) with $(X, Y) \sim \pi$, we use the *Barycentric projection* $T_\pi : \Omega \to \Omega$ to approximate the transport map $T : \Omega \to \Omega$ to be defined as

$$T_\pi(x_i) = \mathbb{E}_\pi[\, Y \mid X = x_i \,] = \frac{1}{a_i} \sum_{j=1}^{m} \pi_{ij} y_j \tag{4.5.1}$$

With palettes, we perform recolouring procedure via the nearest-pixel assignment

$$x \mapsto T(x_{i^*}) \quad \text{where } i^* = \arg\min_i \|x - x_i\|.$$

For the perceptual reasons, we may use the displacement interpolation $T_\alpha : \Omega \to \Omega$ on the maps T and T_π to keep the results closer to the original source:

$$T_\alpha(x) = (1 - \alpha)\, x + \alpha\, T(x), \quad \alpha \in [0, 1]. \tag{4.5.2}$$

In the experiments we evaluate the metrics at each $\alpha \in \{0.4, 0.6, 0.8, 1.0\}$, so that $\alpha = 1.0$ is the full transport $T_\alpha = T$ and smaller values keep the recolouring progressively closer to the source.
Evaluation metrics. To assess the quality of the results we consider the following evaluation metrics based on visual perception (SSIM, Colourfulness index, Gradient magnitude, Laplacian sharpness) and based on transport map estimation (Monge-Ampère residual, diffuseness)

1. **Structural Similarity Index** (SSIM) is a widely used perceptual metric for image quality assessment. We evaluate it on the L channel of CIELAB, between the recoloured and source images. Let x and y refer to corresponding local windows (of size $w \times w$ with $w = 11$) of the L component; the reported SSIM score is the average of the local SSIM over all window positions across the image:

$$\mathrm{SSIM}_{\mathrm{local}}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{4.5.3}$$

   where the constants $C_1, C_2 > 0$ stabilize the ratio ($C_1 = (k_1 D)^2, C_2 = (k_2 D)^2$ with $k_1 = 0.01, k_2 = 0.03$ and D the dynamic range of the L channel, here $D = 1$ after normalization), μ_z and σ_z are the mean and standard deviation of z, and σ_{xy} is the covariance between x and y. The metric combines luminance (similarity of mean intensity), contrast, and structural similarity (Wang et al., 2004). A high SSIM (up to 1) indicates that the luminance structure of the source image is preserved.

2. **Colourfulness index** (Hasler–Süsstrunk) quantifies the perceived colour vividness and is reported in RGB. Let $rg = R - G$ and $yb = \frac{1}{2}(R + G) - B$. The colourfulness of a single image is

$$\mathrm{CF} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} + 0.3\, \sqrt{\mu_{rg}^2 + \mu_{yb}^2}, \tag{4.5.4}$$

   where σ_z and μ_z are the standard deviation and mean of z. We report the absolute difference in colourfulness between the recoloured and target images: a large difference means oversaturation (increase of colourfulness) or a collapse of saturation (decrease of colourfulness).

3. **Gradient magnitude correlation.** Let $L^{(1)}$ and $L^{(2)}$ be the L components of the recoloured and source images in CIELAB. Compute the spatial gradient magnitudes $g_1(i) = \|\nabla L^{(1)}(i)\|$ and $g_2(i) = \|\nabla L^{(2)}(i)\|$ at pixel locations i, and, with g_1 and g_2 vectorized, the Pearson correlation

$$\mathrm{GradCorr} = \frac{\sum_i (g_1(i) - \overline{g_1})(g_2(i) - \overline{g_2})}{\sqrt{\sum_i (g_1(i) - \overline{g_1})^2}\,\sqrt{\sum_i (g_2(i) - \overline{g_2})^2}}. \tag{4.5.5}$$

   Defined on the luminance component of CIELAB, it probes edge strength and texture contrast, complementing SSIM, which relies mostly on local statistics.

4. **Laplacian sharpness** would allow us to catch the blurring and artificial crispness introduced by the method. We compute the discrete Laplacian ΔL_x on the luminance component and report the difference between sharpness score of recoloured and target images, where the sharpness score we take as

$$\mathrm{Sharp}(x) = \mathrm{Var}(\Delta L_x). \tag{4.5.6}$$

5. **Monge–Ampère residual** was introduced in Section 4.3.1 as a structural-accuracy diagnostic of the recovered map:

$$r(x) = \det\big(\nabla T(x)\big)\,\rho_\nu\big(T(x)\big) - \rho_\mu(x), \tag{4.5.7}$$

   and we report its L^1 aggregate $\|r\|_{L^1}$, half of which equals the fraction of misallocated mass. We reuse the same metric here, applied to the Monge map for the Back-and-Forth solver and to the barycentric projection T_π otherwise.

6. **Diffuseness** measures the conditional variance of the target colour around the barycentric projection (4.5.1). For a coupling π and a source point x_i,

$$\mathrm{Var}(Y \mid X = x_i) = \mathbb{E}[\|Y\|^2 \mid X = x_i] - \|\mathbb{E}[Y \mid X = x_i]\|^2 = \frac{1}{a_i}\sum_{j=1}^{m} \|y_j\|^2\,\pi_{ij} - \|T_\pi(x_i)\|^2,$$

   where $a_i = \sum_{j=1}^{m} \pi_{ij}$ is the mass the plan assigns to x_i (equivalently the source histogram weight μ_i). We summarize a plan by the mass-weighted aggregate

$$D(\pi) = \sum_{i=1}^{n} a_i\,\mathrm{Var}(Y \mid X = x_i).$$

   This is reported only for the Kantorovich-type solvers. Because their plans are not deterministic — the entropic regularization spreads each source colour over several targets — $D(\pi)$ quantifies how far the recovered plan is from a Monge map: it vanishes when the plan is induced by a map (each x_i sends all its mass to a single y_j) and grows as the plan becomes more diffuse, e.g. with increasing Sinkhorn regularization ε.

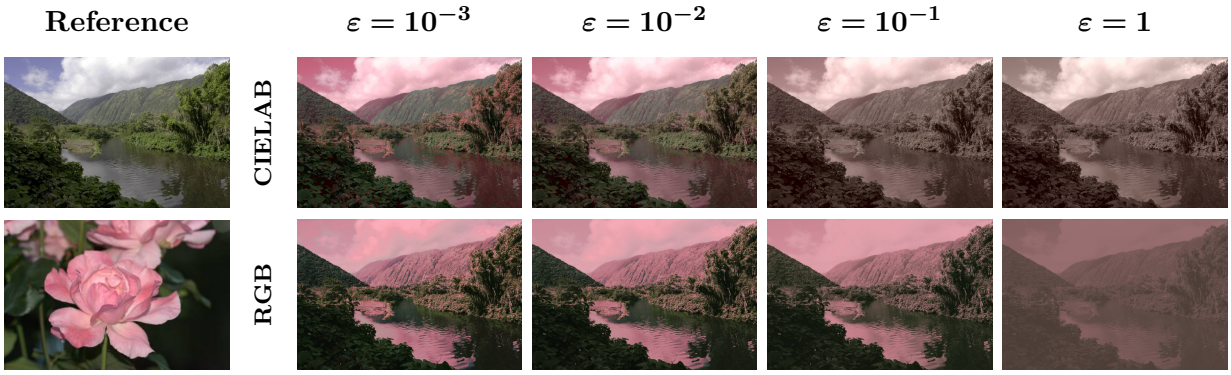


FIGURE 25. ***Visual effects of entropic regularization in Sinkhorn colour transfer (log-domain Sinkhorn).*** *The first row shows the recolouring in CIELAB space and the second row the corresponding results in RGB space; the leftmost column shows the reference source (top) and target (bottom) images. From left to right, the transferred images use regularization levels* $\varepsilon \in \{10^{-3}, 10^{-2}, 10^{-1}, 1\}$. *Increasing* ε *makes the transport plan more diffuse and the output smoother, but also increases the bias of the entropic barycentric estimator* (4.5.1) *of the transport map, which pulls the transferred colours toward the palette mean and washes them out. This is reflected in the growing colourfulness difference* (4.5.4) *between the recoloured and target images as* ε *grows (CIELAB:* 0.0427, 0.0582, 0.1159, 0.1287*; RGB:* 0.0107, 0.0149, 0.0286, 0.1031*).*

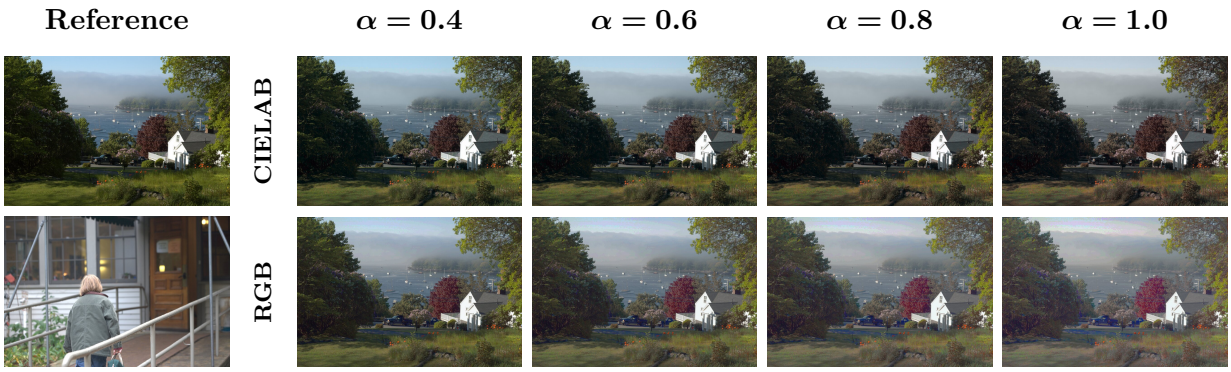


FIGURE 26. ***Displacement interpolation effects for the Back-and-Forth colour transfer map.*** *The first row shows recolouring results in CIELAB space, while the second row shows the corresponding results in RGB space. The first-column images are the source and target references, respectively. From left to right, the transferred images are obtained by applying the displacement interpolation map* $T_\alpha(x)$ *with* $\alpha \in \{0.4, 0.6, 0.8, 1.0\}$. *Increasing* α *strengthens palette matching, reflected by decreasing Sinkhorn divergence in RGB space (*0.15, 0.10, 0.07, 0.06 *for* $\alpha = 0.4, 0.6, 0.8, 1.0$*, respectively), but may reduce structural similarity, as indicated by lower SSIM values in RGB space (*0.87, 0.80, 0.73, 0.67*). This illustrates a trade-off between colour matching and structure preservation.*

Finally, to assess the quality of the resulting histogram on the recoloured image we consider the *Sinkhorn divergence* (2.6.2) between the (empirical) histogram $\hat{\mu}$ of the recoloured image and the target histogram ν, i.e.

$$S_\varepsilon(\hat{\mu}, \nu) = \mathcal{K}^\star_\varepsilon(\hat{\mu}, \nu) - \frac{1}{2}\mathcal{K}^\star_\varepsilon(\hat{\mu}, \hat{\mu}) - \frac{1}{2}\mathcal{K}^\star_\varepsilon(\nu, \nu),$$

with $\varepsilon = 5 \cdot 10^{-4}$. In our experiments, the computation uses at most 50,000 Sinkhorn iterations, tolerance 10^{-5}, and batch size 8192.

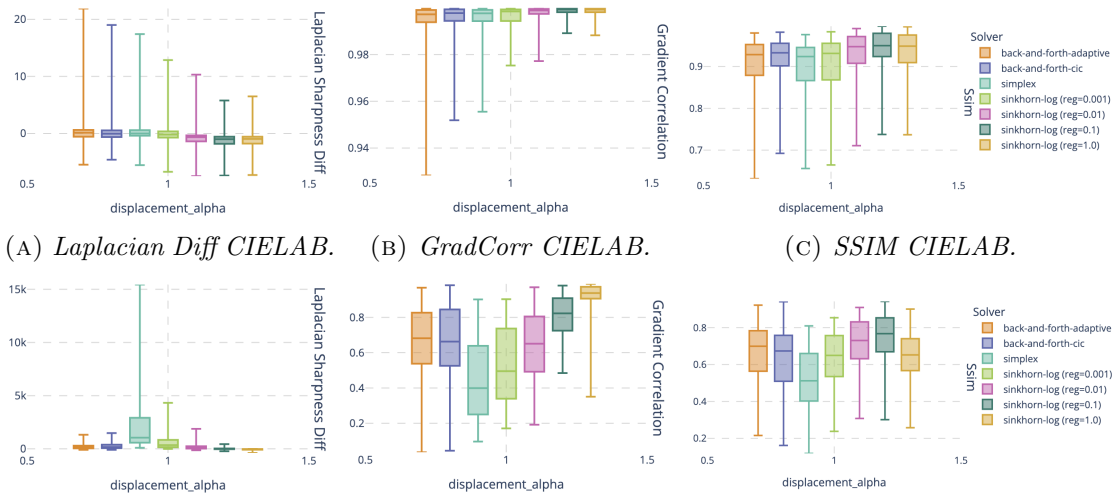
In CIELAB each solver achieves decent results (Figures 28 and 29): the choice of a perceptually

FIGURE 27. ***Comparison of constant cell-like artifacts across solvers.*** *The first row shows the source and target images. The remaining rows compare recolouring results produced by the Back-and-Forth, Sinkhorn, and simplex solvers in CIELAB and RGB colour spaces. Visually, the simplex solver, which computes the exact histogram matching result on the discrete support, produces the strongest piecewise-constant cell-like artifacts. These artifacts are less visible for Sinkhorn and are reduced most clearly for the Back-and-Forth method, whose outputs appear smoother overall.*

α	RGB	CIELAB
0.4	1st: `Back-and-Forth (CIC)` (0.02753) 2nd: *simplex* (0.03134) $\Delta = 0.00381$ (12.2%)	1st: `Back-and-Forth Adaptive` (0.02593) 2nd: *Back-and-Forth (CIC)* (0.02725) $\Delta = 0.00132$ (4.8%)
0.6	1st: `Sinkhorn` ($\varepsilon = 0.001$) (0.02274) 2nd: *simplex* (0.02277) $\Delta = 3.03 \times 10^{-5}$ (0.1%)	1st: `Back-and-Forth Adaptive` (0.01690) 2nd: *simplex* (0.02296) $\Delta = 0.00606$ (26.4%)
0.8	1st: `simplex` (0.01542) 2nd: *Back-and-Forth Adaptive* (0.01616) $\Delta = 0.00074$ (4.6%)	1st: `Back-and-Forth Adaptive` (0.01605) 2nd: *simplex* (0.01717) $\Delta = 0.00112$ (6.5%)
1.0	1st: `simplex` (0.004755) 2nd: *Sinkhorn* ($\varepsilon = 0.001$) (0.01109) $\Delta = 0.00634$ (57.1%)	1st: `simplex` (0.01215) 2nd: *Sinkhorn* ($\varepsilon = 0.001$) (0.01529) $\Delta = 0.00315$ (20.6%)

TABLE 8

Per-configuration winners for the colourfulness metric. *Each cell reports the solver with the smallest median colourfulness error, the second-best solver, and their corresponding median values for the given configuration (colour space, displacement level α). The reported margin is $\Delta = Y_{(2)} - Y_{(1)}$, shown in absolute and relative terms, where Y_s denotes the median colourfulness difference for solver s and $Y_{(1)} \leq Y_{(2)}$ are ordered values across all solvers.*



(A) *Laplacian Diff CIELAB.* (B) *GradCorr CIELAB.* (C) *SSIM CIELAB.*

(D) *Laplacian Diff RGB.* (E) *GradCorr RGB.* (F) *SSIM RGB.*

FIGURE 28. ***Structure preservation metrics across all solvers and colour spaces.*** *Top: CIELAB. Bottom: RGB. Across all three metrics, the distribution of CIELAB is generally more concentrated than in RGB, indicating a more stable structure-preserving behaviour, which is not surprising, since the transfer is applied only to the chromatic channels A and B, while the luminance channel L is left unchanged. By contrast, RGB is more variable, especially for Laplacian difference and gradient correlation, since the transfer acts directly on intensity-carrying channels — simplex method alters edges and introduces artificial sharpness due to the exact histogram matching, while the* ***Back-and-Forth method shows better statistics*** *in edges preservation overall.*

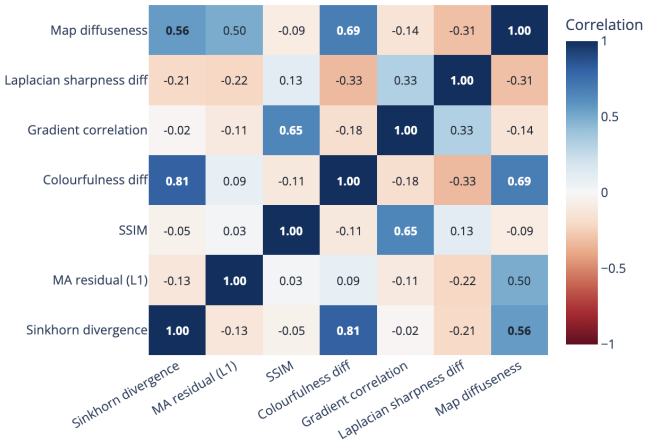
well-behaved space does a large part of the job. Moreover, since in CIELAB we leave the luminance channel untouched, we can retain the same brightness as the original image while still shifting the

palette towards that of the target (Figure 26). In RGB, by contrast, the all-channel transfer alters brightness, which may be undesirable as it can distort the perception of the image.

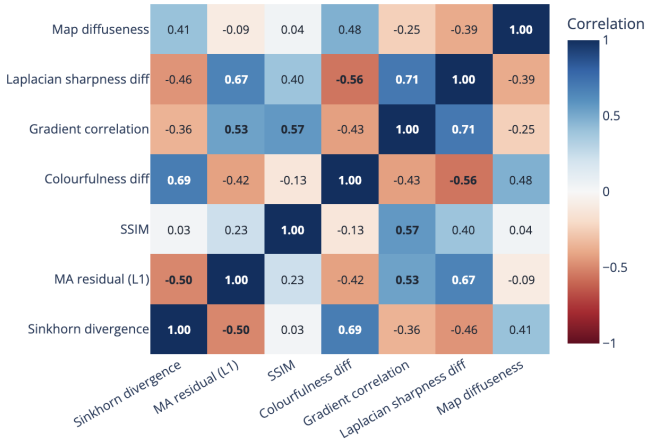
Turning to RGB reveals more interesting behaviour: even at full transport the metrics trade off against one another, with stronger colour matching coming at the cost of structure preservation (Figure 29). Simplex, as a representative of exact Kantorovich solvers, requires a barycentric projection T_π (4.5.1) to be used for colour transfer, and this combination produces piecewise-constant, cell-like artifacts (Figure 27) that are visible in the structure-preservation metrics: the simplex method alters edges and introduces artificial sharpness due to the exact histogram matching (Figure 28, Table 8). Simplex may therefore not be the method of choice here. Sinkhorn, by contrast, does not introduce such strong effects thanks to its regularization, but exhibits a trade-off between colour preservation and structure preservation; choosing a suitable regularization level can strike a good balance (Figure 25, Figure 27, Figure 28, and Table 8).

Back-and-Forth may be an alternative to Sinkhorn here: by computing a more accurate, less diffuse transport map, i.e. a smaller Monge–Ampère residual, it sits on the structure-preserving side of the trade-off, yielding higher gradient correlation and a smaller Laplacian sharpness difference, hence fewer oversmoothing and edge artifacts, while losing little in colourfulness (Figure 28, Figure 29, Table 8).

Overall, both Sinkhorn and Back-and-Forth stand out as the most suitable solvers for colour transfer in RGB: Sinkhorn offers a tunable colour–structure trade-off through its regularization, while Back-and-Forth provides a strong, less diffuse alternative that preserves structure well without losing much in colourfulness.



(A) *CIELAB. Colourfulness matching improves together with palette matching — its correlation with the Sinkhorn divergence is the largest off-diagonal entry — and with a less diffuse transport plan. The Monge–Ampère residual is uncorrelated with everything except map diffuseness, so for chroma-only transfer it is not an informative predictor of perceptual quality. SSIM and gradient correlation are positively correlated, as expected for two descriptors of the same luminance structure.*



(B) *RGB. A colour-versus-structure trade-off appears even at full transport: stronger palette and colourfulness matching coincide with weaker edge and sharpness preservation (the colourfulness–Laplacian and Sinkhorn–gradient-correlation entries are negative). A more accurate transport map, i.e. a smaller Monge–Ampère residual, goes together with higher gradient correlation and a smaller Laplacian sharpness difference, consistent with fewer oversmoothing and edge artifacts, and the three structure metrics are mutually positively correlated. Map diffuseness sits on the colour side of this trade-off rather than the structure side: a more diffuse plan matches the palette and colourfulness worse but preserves edges and sharpness better. This is consistent with Figure 25, where larger entropic regularization raises both diffuseness and the colourfulness change, and with Figure 27, where the sharpest (least diffuse) simplex plan produces the strongest piecewise-constant artifacts.*

FIGURE 29. ***Spearman correlation between the colour transfer metrics*** *computed over all runs with* $\alpha = 1.0$ *and regularization* $\varepsilon = 10^{-3}$ *for Sinkhorn. So that the sign of every entry reads directly as agreement in quality, the five metrics for which lower is better — Sinkhorn divergence, Monge–Ampère residual (L1), map diffuseness, colourfulness difference, and Laplacian sharpness difference — are negated before the correlation is computed; for all seven metrics a larger value then means better performance, as it already does for SSIM and gradient correlation. A positive entry therefore indicates that the two qualities improve together, a negative entry a trade-off.*

Part 5

Discussion and Research Directions

Summary. In this thesis we carried out a comparative and experimental study of computational methods for discrete optimal transport under the squared-Euclidean cost. We benchmarked the exact simplex method, the entropic Sinkhorn algorithm in its standard and log-domain forms, first-order dual ascent (plain gradient ascent, SGD with momentum, and Adam), the quasi-Newton L-BFGS method, the chi-square regularized PDHG of Ruban and Gerolin (Ruban and Gerolin, 2025), and the Back-and-Forth method of Jacobs and Léger (Jacobs and Léger, 2020). The comparison spanned dimensions $d \in \{1, 2, 3\}$, a range of grid resolutions, several regularization levels, and a collection of distribution families chosen to cover different modality, tail thickness, skewness, and anisotropy. We assessed the solvers against three criteria — stability, runtime, and accuracy — and validated the conclusions on a real application, colour transfer.

Main findings. Across the synthetic benchmark, log-domain Sinkhorn was the most robust and scalable solver. The price of this robustness is the entropic bias: at a fixed ε the recovered cost does not improve with grid refinement, and driving $\varepsilon \to 0$ to reduce the bias requires an iteration budget that grows like $O(\log(1/\delta)/\varepsilon)$ (Section 3.2, Table 3). The exact simplex solver recovers the analytic cost up to discretization but does not scale: it increasingly hits the iteration cap as the grid is refined, and becomes impractical in 2D and 3D. The first-order and L-BFGS dual solvers were the least competitive, with convergence problems on fine grids and at small ε. PDHG with chi-square regularization behaves differently from the entropic solvers — its bias stays stable as $\varepsilon \to 0$ — but its memory cost made it impractical beyond 1D.

Back-and-Forth. The Back-and-Forth method occupies a distinct niche: it is the only solver in the comparison that returns a transport map directly, rather than a coupling. Its structural output is sound. The cost underestimation it exhibits is a discretization artifact of the map-based cost, not an algorithmic failure, and it shrinks under grid refinement. In 2D and 3D the Cloud-in-Cell scheme scales better than simplex, which makes Back-and-Forth a practical map-based alternative once the grid is fine.

The "best" solver is application-dependent. On the synthetic benchmark, accuracy is measured against a known reference (the LP cost, or the analytic map on Gaussian instances), so the exact simplex solver is the natural gold standard. Colour transfer shows this is not the whole story: in RGB the exact histogram matching of simplex produces piecewise-constant artifacts and artificial sharpness, while a slightly biased but smoother map — from Sinkhorn at a suitable ε, or natively from Back-and-Forth — preserves structure better and loses little in colour matching. When the source and target distributions differ in nature, matching them exactly is not always desirable. A synthetic benchmark measures fidelity to an exact reference and cannot, on its own, capture such task-specific criteria, which is why we paired it with a concrete application.

Why Sinkhorn remains the default. Our results also help explain why the Sinkhorn algorithm has remained the method of choice despite its bias. It is robust — in log-domain form it is numerically stable down to small ε and converges reliably across problem sizes — and it is remarkably easy to modify and extend. A large body of work builds directly on it: greedy coordinate-descent variants such as Greenkhorn that reduce its cost (Altschuler et al., 2017); stabilized and ε-scaling implementations that make it usable at small regularization (Schmitzer, 2019); generalizations to unbalanced transport (Chizat et al., 2018); and debiased variants such as Sinkhorn divergences (Feydy et al., 2019). This combination of robustness and modifiability, rather than accuracy alone, is a large part of its popularity.

A new method for Wasserstein barycenters. Beyond the comparison study, the thesis in-

troduced a new computational method for the (outer-)regularized 2-Wasserstein barycenter (Section 4.4). Instead of regularizing the transport plans, the method regularizes the dual objective and performs a Sobolev gradient ascent on the dual functionals combined with the Back-and-Forth scheme. The outer regularization enforces absolute continuity of the barycenter and damps the grid-scale noise of the plain Back-and-Forth map, while keeping the maps sharper than convolutional Sinkhorn. To our knowledge this is a new way to combine the Back-and-Forth idea with an outer regularization, and it suggests a broader research program, described next.

5.1 Limitations

Our experiments were performed in low dimensions ($d \leq 3$), so the conclusions drawn here do not directly transfer to the high-dimensional, sample-based regime where much of machine-learning optimal transport lives. Most of the study uses the squared Euclidean cost (or a similar quadratic cost); other costs are untested, and the BaFFO barycenter is scoped to separable costs. Across the benchmark the reference is the exact (simplex) LP solution, which is itself reliable only where simplex converges; the Gaussian closed form (Takatsu, 2011) is used separately, as an independent sanity check on selected instances. This is a genuine limitation: on fine grids in two and three dimensions simplex increasingly hits the iteration cap, so the reference against which accuracy is measured becomes unreliable at exactly the resolutions of interest.

Colour transfer and Wasserstein barycenters are the only real-world examples. A synthetic benchmark measures fidelity to a known reference and cannot, on its own, capture task-specific quality criteria (Section 5.2).

Some limitations are practical. The chi-square-regularized PDHG solver could not be run beyond one dimension because of its memory cost, so that comparison is limited to 1D. All conclusions are conditional on the tested hyperparameter ranges and on a fixed iteration budget, which makes the reported hit-rates budget-relative. Runtimes were measured with a single JAX implementation on one GPU and are therefore indicative rather than absolute; a different implementation could reorder the rankings.

Finally, the barycenter method of Section 4.4 is presented at a formal level: the duality and first-variation computations are not made rigorous, and a convergence analysis of the scheme remains open. Convergence of the underlying Back-and-Forth iteration is not established even in the classical case (Jacobs and Léger, 2020).

5.2 Research directions

Analysis of the Back-and-Forth method. The theoretical understanding of Back-and-Forth is still incomplete. The monotonicity of the scheme is only guaranteed conditionally, under a uniform Hessian bound on the iterates (Jacobs and Léger, 2020) ; a convergence proof without such an assumption is, to our knowledge, not available, and the existing analysis of the classical method does not extend directly to our outer-regularized and barycenter variants. Establishing convergence for these generalized schemes is the most important open problem raised by this work.

From barycenters to a general framework. The outer-regularized barycenter method is the simplest case of the more general *Back-and-Forth Flow* (BaFFO) framework that we are developing in joint work with D. Ruban and A. Gerolin (Zhytkevych et al., 2026). There one places a measure at each node of a graph, an optimal-transport cost on each edge, and an outer-regularizing functional on the nodes, and runs the same dual ascent. This framework covers, besides barycenters, variational mean-field games and trajectory inference. A first version has been submitted to NeurIPS 2026; we are currently working on a rigorous justification of the duality and convergence

statements, which are given here only at a formal level.

Faster primitives and higher dimensions. The per-iteration cost of Back-and-Forth is dominated by the c-transform and the pushforward. Our c-transform uses the Convex Hull Trick, which is $O(n)$ per line but still relies on exact Legendre-type transforms; faster GPU implementations would improve scalability further. The method is also currently grid-based, which limits it to low dimensions; extending it to sample-based settings — for instance with neural-network parameterizations of the dual potentials and the Sobolev gradients — is a natural next step toward high-dimensional problems.

Application-driven comparison. Finally, as the colour-transfer study showed, the value of a solver is best judged on a concrete task. A useful direction is to extend the comparison to further applications — beyond colour transfer and barycenters — where the structure of the problem, rather than fidelity to an exact reference, decides which method is preferable. Just as the flexibility of Sinkhorn drove much of its adoption, we expect that reconsidering and modifying the Back-and-Forth method for new problem classes — as we did for barycenters — is where it can contribute most.

APPENDICES

A Theory

A.1 The linear-programming view of OT and the simplex method

The primal problem (3.0.3) is an instance of a linear programming (LP) problem. To recast it in canonical form we vectorise the matrices: set $c = -\vec{(}C) \in \mathbb{R}^{nm}$ and $\gamma = \vec{(}\pi) \in \mathbb{R}^{nm}_{0,+}$. The primal then reads

$$\max_{\gamma \in \mathbb{R}^{nm}_{0,+}} c^\top \gamma, \tag{A.1}$$

subject to

$$A\,\gamma = p := \begin{pmatrix} a \\ b \end{pmatrix}, \tag{A.2}$$

$$\gamma \geq 0, \tag{A.3}$$

where A is the $(n+m) \times nm$ matrix whose top block $I_n \otimes \mathbf{1}_m^\top$ sums each row of π and whose bottom block $\mathbf{1}_n^\top \otimes I_m$ sums each column:

$$A = \begin{bmatrix} I_n \otimes \mathbf{1}_m^\top \\ \mathbf{1}_n^\top \otimes I_m \end{bmatrix}.$$

The $n+m$ marginal equations are not independent: summing the n row equations and summing the m column equations both give the total-mass equation $\sum_{i,j} \pi_{ij} = 1$, so exactly one equation is redundant and $\operatorname{rank}(A) = n + m - 1 =: r$. We delete one redundant equation, leaving a system of full row rank r. By the rank–nullity theorem, since $A\gamma = p$ is consistent there exists an index set $B = \{j_1, \ldots, j_r\} \subset [nm]$ whose columns $A_{[j_1]}, \ldots, A_{[j_r]}$ are linearly independent and span p, so the corresponding basic solution has at most r nonzero entries. This partitions the variable indices into *basic* and *non-basic* sets B and N:

$$[nm] = B \cup N, \quad B \cap N = \varnothing, \tag{A.4}$$

$$\gamma_{[B]} = (\gamma_j : j \in B) \in \mathbb{R}^r_{0,+}, \tag{A.5}$$

$$\gamma_{[N]} = (\gamma_j : j \in N) \in \mathbb{R}^{nm-r}_{0,+}. \tag{A.6}$$

The feasible set $\{\gamma \in \mathbb{R}^{nm}_{0,+} : A\gamma = p\}$ is a *bounded* polyhedron of dimension $nm - r = (n-1)(m-1)$: every entry obeys $0 \leq \pi_{ij} \leq \min(a_i, b_j)$, so no coordinate can run to infinity. It contains infinitely many points but finitely many extreme points, and these form the search space of the simplex method. The extreme points are the **basic feasible solutions** (BFS); each has at most $r = n + m - 1$ positive entries and at least $nm - r$ zero entries, with exactly r positive entries in the nondegenerate case.

Iterating over the basic feasible solutions is what defines the *simplex method*. As an iterative method it needs a starting point, and since the search space is constrained this point must itself be a BFS. The method therefore proceeds in two phases: **Phase I**, which finds an initial BFS (or

detects infeasibility); and **Phase II**, which iterates over BFS to reach the optimal one. We stop once no move to an adjacent BFS lowers the cost.

Pivoting. Pivoting is the mechanism by which the simplex method moves between bases and updates its working representation of the system (the *tableau*); the original matrix A is unchanged. The variable whose column we pivot on enters the basis (*entering variable*) and a second variable leaves it (*leaving variable*), preserving the BFS structure (r basic and $nm - r$ non-basic variables). Writing A for the current tableau, the *pivot* on element a_{ij} produces the tableau $\overline{A}$ obtained by the row operation

$$\overline{A}_{[k]} = \begin{cases} \dfrac{1}{a_{ij}} A_{[i]} & \text{if } k = i, \\ A_{[k]} - \dfrac{a_{kj}}{a_{ij}} A_{[i]} & \text{if } k \neq i, \end{cases} \qquad 1 \leq k \leq r. \tag{A.7}$$

After this operation $\overline{a}_{ij} = 1$ and every other entry of column j is zero, so with the non-basic variables set to zero, $\overline{a}_{ij}$ is the only term contributing to equation i, and γ_{ij} is set to $\overline{p}_i$ to satisfy $\overline{A}_{[i]}\gamma = \overline{p}_i$. In matrix form this is the left-multiplication of A by

$$P = \begin{pmatrix} 1 & & -a_{1j}/a_{ij} & & \\ & \ddots & \vdots & & \\ & 1 & -a_{i-1,j}/a_{ij} & & \\ & & 1/a_{ij} & & \\ & & -a_{i+1,j}/a_{ij} & 1 & \\ & & \vdots & & \ddots \\ & & -a_{r,j}/a_{ij} & & 1 \end{pmatrix}, \tag{A.8}$$

applied to both sides: $PA\gamma = Pp$.

Phase I. This phase concerns only feasibility: finding a BFS or proving none exists. (For balanced OT a feasible BFS is in fact available directly, e.g. by the northwest-corner construction, so Phase I is needed only in the generic LP setting.)

1. Introduce artificial variables $\{\gamma_i^a : i \in [r]\}$ to create an obvious initial basis, requiring that in the final BFS of this phase they are all non-basic (the basis consists only of original variables):

$$\begin{array}{l} \max z = c^\top \gamma \\ \text{s.t. } A\gamma = p,\ \gamma \geq 0 \end{array} \xrightarrow[\text{add } \{\gamma_i^a\}_{i\in[r]}]{} \begin{array}{l} \min w = \displaystyle\sum_{i=1}^{r} \gamma_i^a \\ \text{s.t. } A\gamma + I_r\gamma^a = p,\ \gamma \geq 0,\ \gamma^a \geq 0. \end{array}$$

   Replacing $\min w$ by $\max w' = -w$ and using $\gamma^a = p - A\gamma$ to express the objective in the original variables:

$$\max w' = -\mathbf{1}_r^\top p + (\mathbf{1}_r^\top A)\gamma \quad \text{s.t. } A\gamma + I_r\gamma^a = p,\ \gamma \geq 0,\ \gamma^a \geq 0.$$

2. Apply the Phase II procedure to this auxiliary problem. There are three outcomes:

   (a) $w' = 0$ and no artificial variable is basic: delete the artificial columns and start Phase II from the obtained BFS.

   (b) $w' < 0$ at optimality (so $w > 0$): the original system is infeasible.

(c) $w' = 0$ but some artificial variable remains basic: pivot to exchange each such artificial variable with an original one, reducing to case (a).

Phase II.

1. To pivot conveniently, treat the objective as a constraint: with a variable z, the objective row is $-z + c^\top \gamma = -c_0$. Denote the current (updated) tableau matrices by $\overline{A}, \overline{c}, \overline{p}$. The initial tableau is

$$\begin{array}{c|cc|c} -z & \gamma_{[B]} & \gamma_{[N]} & \\ 1 & 0 & (c_{[N]})^\top & -c_0 \\ \hline & A_{[B]} = I_r & A_{[N]} & p \end{array},$$

where $A\gamma = p$ is written $A_{[B]}\gamma_{[B]} + A_{[N]}\gamma_{[N]} = p$.

2. If the reduced cost $\overline{c}_k \le 0$ for all $k \in N$, the current BFS is optimal — no non-basic variable can increase the objective — and we stop.

3. Otherwise find the entering and leaving variables:

   (a) Pick a column j with reduced cost $\overline{c}_j > 0$; then γ_j is the *entering variable*. Choosing it carelessly can cause cycling (revisiting a BFS), so an anti-cycling rule such as *Bland's rule* picks the smallest such index j.

   (b) Unboundedness: if $\overline{a}_{kj} \le 0$ for *all* k, the column imposes no upper bound on γ_j, so $\gamma_j \to \infty$ and $z \to \infty$; the LP is unbounded and we stop. (This cannot occur for balanced OT, whose feasible set is bounded.)

   (c) Min-ratio test. The basic variables update as

$$\gamma_{[B]}(\theta) = \overline{p} - \overline{a}_j\,\theta, \qquad \overline{p} = A_{[B]}^{-1}p, \quad \overline{a}_j = A_{[B]}^{-1}A_{[j]}, \quad \theta = \gamma_j. \tag{A.9}$$

   Increasing $\theta = \gamma_j$ from zero, the first basic variable to reach zero leaves. The pivot row i is the one attaining

$$\frac{\overline{p}_i}{\overline{a}_{ij}} = \min_{k:\,\overline{a}_{kj}>0} \frac{\overline{p}_k}{\overline{a}_{kj}},$$

   and the basic variable γ_l in that row, with $l = B(i)$, is the *leaving variable*. Bland's rule breaks ties by smallest index.

4. Pivot on $\overline{a}_{ij}$, which updates the basic and non-basic sets,

$$B \leftarrow (B \setminus \{l\}) \cup \{j\}, \qquad N \leftarrow (N \setminus \{j\}) \cup \{l\},$$

   and corresponds to left-multiplication by $P = (A_{[B]})^{-1}$:

$$PA_{[B]} = \overline{A}_{[B]} = I_r, \quad PA_{[N]} = \overline{A}_{[N]} = A_{[B]}^{-1}A_{[N]}, \quad Pp = \overline{p} = A_{[B]}^{-1}p.$$

   Treating the objective as a constraint, the current and original objective values agree; substituting $\gamma_{[B]} = \overline{p} - \overline{A}_{[N]}\gamma_{[N]}$ gives

$$c_{[B]}^\top \gamma_{[B]} + c_{[N]}^\top \gamma_{[N]} = \overline{c}_0 + \overline{c}_{[N]}^\top \gamma_{[N]},$$

with the updated blocks

$$\overline{c}_{[N]}^{\top} = c_{[N]}^{\top} - c_{[B]}^{\top} A_{[B]}^{-1} A_{[N]}, \qquad \overline{c}_0 = c_{[B]}^{\top} A_{[B]}^{-1} p.$$

The updated tableau is

$$\begin{array}{c|c|cc|c} -z & \gamma_{[B]} & & \gamma_{[N]} & \\ 1 & 0 & c_{[N]}^{\top} - c_{[B]}^{\top} A_{[B]}^{-1} A_{[N]} & & -c_{[B]}^{\top} A_{[B]}^{-1} p \\ \hline & I_r & & A_{[B]}^{-1} A_{[N]} & A_{[B]}^{-1} p \end{array}.$$

Return to step 2.

The two-phase construction and revised-simplex notation above follow (Goemans, 2015).

A.2 Sion's minimax theorem

Theorem A.1 (Sion's minimax theorem)**.** *Let X and Y be convex sets in linear topological spaces, and assume one of them is compact. If $f : X \times Y \to \mathbb{R}$ is such that, for every fixed $y \in Y$, $f(\cdot, y)$ is upper semicontinuous and quasi-concave on X, and for every fixed $x \in X$, $f(x, \cdot)$ is lower semicontinuous and quasi-convex on Y, then*

$$\sup_{x \in X} \inf_{y \in Y} f(x, y) = \inf_{y \in Y} \sup_{x \in X} f(x, y).$$

This is Corollary 3.3 in (Sion, 1958).

A.3 Cost underestimation by the barycentric map

Remark A.2 (Cost underestimation by the barycentric map). Let π be a transport plan with first marginal μ, and let T_π be the barycentric map it induces, $T_\pi(x_i) := \frac{1}{\mu_i} \sum_j \pi_{ij}\, y_j$. Then the cost of T_π does not exceed the cost of π:

$$\sum_{i=1}^{n} \|x_i - T_\pi(x_i)\|^2 \, \mu_i \;\le\; \sum_{i=1}^{n} \sum_{j=1}^{m} \|x_i - y_j\|^2 \, \pi_{ij}. \tag{A.10}$$

Proof. Fix x_i with $\mu_i > 0$ and set $\lambda_{ij} := \pi_{ij}/\mu_i$. The marginal constraint $\sum_j \pi_{ij} = \mu_i$ gives $\sum_j \lambda_{ij} = 1$ with $\lambda_{ij} \ge 0$, so $(\lambda_{ij})_j$ is a probability distribution over the y_j and $T_\pi(x_i) = \sum_j \lambda_{ij} y_j$ is their barycenter. Applying Jensen's inequality to the convex function $f(z) = \|x_i - z\|^2$,

$$\|x_i - T_\pi(x_i)\|^2 = f\Big(\textstyle\sum_j \lambda_{ij} y_j\Big) \le \sum_j \lambda_{ij}\, f(y_j) = \sum_j \lambda_{ij}\, \|x_i - y_j\|^2. \tag{A.11}$$

Multiplying by μ_i and summing over i yields (A.10). □

A.4 Discretization floor of the Monge–Ampère residual

Origin of the floor

The residual $r(x) = \det(\nabla T(x))\, \rho_\nu(T(x)) - \rho_\mu(x)$ vanishes identically for the exact map in the continuum. On a finite grid, however, $T(x)$ generally does not land on a grid node, so the target density $\rho_\nu(T(x))$ must be *interpolated* from its values at the surrounding nodes. For multilinear interpolation of a C^2 function f, the pointwise error at an off-node location averages, over a cell,

to

$$\widehat{f}(y) - f(y) \;\approx\; \frac{h^2}{12}\,\Delta f(y), \qquad \Delta f \;=\; \sum_{i=1}^{d} \partial_i^2 f, \tag{A.12}$$

where the factor $\frac{1}{12}$ is the cell-averaged coefficient of the second-order Taylor remainder and Δ is the Laplacian. Substituting $f = \rho_\nu$ at $y = T(x)$, the computed residual splits as

$$r_{\mathrm{comp}}(x) \;=\; \underbrace{r_{\mathrm{exact}}(x)}_{=0} \;+\; \det(\nabla T(x))\,\big[\widehat{\rho_\nu} - \rho_\nu\big](T(x)) \;\approx\; \frac{h^2}{12}\,\det(\nabla T(x))\,\Delta\rho_\nu(T(x)). \tag{A.13}$$

Aggregating in L^1 and pushing forward to the target domain via the change of variables $y = T(x)$, $\det(\nabla T)\,\mathrm{d}x = \mathrm{d}y$,

$$\|r\|_{L^1} \;\approx\; \frac{h^2}{12}\int \big|\Delta\rho_\nu(y)\big|\,\mathrm{d}y. \tag{A.14}$$

The floor is therefore set by the total absolute curvature of the target density.

Gaussian targets

For an isotropic Gaussian $\rho_\nu = \mathcal{N}(m_\nu, \sigma_\nu^2 I_d)$ the Laplacian is available in closed form. Writing $\rho := \rho_\nu$,

$$\partial_i\rho = \rho\left(-\frac{y_i - m_{\nu,i}}{\sigma_\nu^2}\right), \qquad \partial_i^2\rho = \rho\left(\frac{(y_i - m_{\nu,i})^2}{\sigma_\nu^4} - \frac{1}{\sigma_\nu^2}\right), \tag{A.15}$$

so that, summing over i,

$$\Delta\rho(y) = \rho(y)\left(\frac{\|y - m_\nu\|^2}{\sigma_\nu^4} - \frac{d}{\sigma_\nu^2}\right) = \frac{\rho(y)}{\sigma_\nu^2}\left(\frac{\|y - m_\nu\|^2}{\sigma_\nu^2} - d\right). \tag{A.16}$$

The integral in (A.14) is then an expectation under ρ_ν itself:

$$\int |\Delta\rho(y)|\,\mathrm{d}y = \frac{1}{\sigma_\nu^2}\,\mathbb{E}_{Y\sim\rho_\nu}\left[\left|\frac{\|Y - m_\nu\|^2}{\sigma_\nu^2} - d\right|\right]. \tag{A.17}$$

Under $Y \sim \mathcal{N}(m_\nu, \sigma_\nu^2 I_d)$ the standardized coordinates $Z_i = (Y_i - m_{\nu,i})/\sigma_\nu$ are i.i.d. $\mathcal{N}(0,1)$, so $\|Y - m_\nu\|^2/\sigma_\nu^2 = \sum_{i=1}^d Z_i^2 \sim \chi^2(d)$. Hence

$$\boxed{\|r\|_{L^1} \;\approx\; \frac{h^2}{12}\,\frac{\mathbb{E}\big[\,|\chi^2(d) - d|\,\big]}{\sigma_\nu^2}} \tag{A.18}$$

which is the discretization floor used in the main text.

Growth with dimension

The dimensional dependence enters only through $g(d) := \mathbb{E}\,|\chi^2(d) - d|$, the mean absolute deviation of a chi-squared variable from its mean d. Since $\mathrm{Var}(\chi^2(d)) = 2d$, a central-limit argument gives $g(d) \sim \sqrt{2d}\cdot\mathbb{E}|\mathcal{N}(0,1)| = \sqrt{4d/\pi}$ for large d: the floor grows only as $\sqrt{d}$, sublinearly, not as a factor of d. Numerically $g(1) \approx 0.97$, $g(2) \approx 1.47$, $g(3) \approx 1.85$, in ratios $1 : 1.52 : 1.91$ relative to $d = 1$. Combined with the coarser grids used in higher dimension (larger h), this explains the bulk of the residual increase from 1D to 3D observed in Table 7.

Numerical confirmation

Replacing the interpolated ρ_ν with the exact continuous density in the residual computation drops the measured value from $\sim 10^{-3}$ to $\sim 10^{-7}$ (floating-point noise), confirming that the floor is entirely an interpolation artifact. Halving h (doubling the grid resolution $N \to 2N$) divides the measured $\|r\|_{L^1}$ by 4.00 to two digits, confirming the $O(h^2)$ rate of (A.12). Finally, evaluating (A.18) directly against the measured residual for $\mathcal{N}(m_\nu, \sigma_\nu^2 I_d)$ targets agrees to within 2–5% across $d = 1, 2, 3$, the small over-prediction being the $O(h^4)$ correction and the discrete-sum approximation of the integral, both vanishing as $N \to \infty$.

B Computational algorithms and pseudocode

B.1 Convex hull trick

Algorithm 4 Offline Convex Hull Trick (Upper Envelope of Lines)

Input: $(x_1, \ldots, x_n)$, $b_i = f(x_i)$
Helper: intersection of two lines (m_1, b_1), (m_2, b_2) is $x = \frac{b_2 - b_1}{m_1 - m_2}$
Sort `LINES` by increasing slope m_i, breaking ties by b_i
`HULL` $\leftarrow$ empty stack of lines
`BP` $\leftarrow$ empty stack of breakpoints
for each line $(m_{\text{new}}, b_{\text{new}})$ in `LINES` **do**
    while `HULL` not empty and last slope $= m_{\text{new}}$ **do**
        if $b_{\text{new}} \leq$ last intercept of `HULL` **then**
            skip this line (useless)
        else
            pop last line from `HULL`; if `BP` not empty, pop last breakpoint
        end if
    end while
    while `HULL` has at least 2 lines **do**
        $x_{\text{prev}} \leftarrow$ last element of `BP`
        $\ell_{\text{prev1}} \leftarrow$ second-to-last line in `HULL`
        $\ell_{\text{prev2}} \leftarrow$ last line in `HULL`
        if intersect$(\ell_{\text{prev2}}, \ell_{\text{prev1}}) \geq$ intersect$(\ell_{\text{prev2}}, (m_{\text{new}}, b_{\text{new}}))$ **then**
            pop last line from `HULL`; pop last element from `BP`
        else
            break
        end if
    end while
    if `HULL` empty **then**
        push $(m_{\text{new}}, b_{\text{new}})$ to `HULL`
    else
        $x_{\text{start}} \leftarrow$ intersect(last line of HULL, $(m_{\text{new}}, b_{\text{new}})$)
        push $(m_{\text{new}}, b_{\text{new}})$ to `HULL`
        push x_{start} to `BP`
    end if
end for

B.2 Cloud-in-Cell (CIC) pushforward

Each cell centre x_i is treated as a "cloud" whose mass is redistributed onto the surrounding 2^d cell centres with multilinear weights; this is the first-order (cloud-in-cell) particle–mesh assignment scheme (Deserno and Holm, 1998; Hockney and Eastwood, 1988).

We work on a uniform grid on $[0,1]^d$, where d is the dimension and $\mathbf{n} = (n_1, \dots, n_d)$ is the number of cells along each axis. Cell indices are $i = (i_1, \dots, i_d) \in \mathcal{I} := \prod_{k=1}^{d} \{0, \dots, n_k - 1\}$, with cell-centre positions

$$x_i = \left(\frac{i_1 + 1/2}{n_1}, \dots, \frac{i_d + 1/2}{n_d} \right) \in [0,1]^d,$$

step sizes $h_k = 1/n_k$, and the source density and potential sampled at centres,

$$\rho_i \approx \rho(x_i), \qquad \psi_i \approx \psi(x_i), \qquad i \in \mathcal{I}.$$

The pushforward is computed in several steps. First, we approximate the gradient by central differences,

$$\left(\frac{\partial \psi}{\partial x_k} \right)_i \approx \frac{\psi_{i+e_k} - \psi_{i-e_k}}{2h_k}, \qquad 1 \le i_k \le n_k - 2,$$

where e_k is the unit vector in direction k, with one-sided differences at the boundaries,

$$\left(\frac{\partial \psi}{\partial x_k} \right)_i \approx \frac{\psi_{i+e_k} - \psi_i}{h_k} \quad (i_k = 0), \qquad \left(\frac{\partial \psi}{\partial x_k} \right)_i \approx \frac{\psi_i - \psi_{i-e_k}}{h_k} \quad (i_k = n_k - 1).$$

The image $y_i := T(x_i) \approx x_i + \nabla_h \psi(x_i)$ corresponds to the continuous cell-centre index

$$s_{i,k} := i_k + n_k \left(\frac{\partial \psi}{\partial x_k} \right)_i \qquad \big(= n_k \, y_{i,k} - \tfrac{1}{2} \big),$$

which we clamp to the domain,

$$\tilde{s}_{i,k} := \min\big(\max(s_{i,k}, 0), n_k - 1 \big),$$

and split into a base index and a fractional part,

$$b_{i,k} := \lfloor \tilde{s}_{i,k} \rfloor, \qquad f_{i,k} := \tilde{s}_{i,k} - b_{i,k} \in [0,1).$$

For each $c = (c_1, \dots, c_d) \in \{0,1\}^d$ define the target index $j(i,c) := b_i + c$, with k-th component $j_k(i,c) = b_{i,k} + c_k$, the per-axis weight

$$w_{i,k}(c_k) := \begin{cases} 1 - f_{i,k}, & c_k = 0, \\ f_{i,k}, & c_k = 1, \end{cases}$$

and the full weight $w_i(c) := \prod_{k=1}^{d} w_{i,k}(c_k)$. The pushed-forward density is then

$$\rho'_j := \sum_{i \in \mathcal{I}} \sum_{c \in \{0,1\}^d} \rho_i \, w_i(c) \, \mathbf{1}_{\{j = j(i,c)\}}.$$

Algorithm 5 CIC pushforward via multilinear interpolation (pseudocode)

Input: potential ψ, density array
Compute $\nabla\psi$ using central (and one-sided) differences
for each grid index i **do**
    $s \leftarrow i + \nabla\psi(i)$ ▷ displaced cell-centre index
    $s_{\text{base}} \leftarrow \lfloor s \rfloor$ (clamped), $f \leftarrow \{s\}$
    for each corner $c \in \{0,1\}^d$ **do**
        $j \leftarrow s_{\text{base}} + c$
        $\omega \leftarrow \prod_{\ell=1}^{d} [\,(1-f_\ell) \text{ if } c_\ell = 0 \text{ else } f_\ell\,]$
        tgt_idx $\leftarrow \sum_{\ell=1}^{d} j_\ell \cdot \text{stride}_\ell$
        out[tgt_idx] += density[i] $\cdot\,\omega$
    end for
end for
Output: out (pushed-forward density field)

B.3 Adaptive pushforward

The objective is to approximate the pushforward measure $T_{\#}\rho$ under the potential ψ. We work in the unit cube $[0,1]^d$ with $\mathbf{n} = (n_1, \ldots, n_d)$ cells along each axis and step $h_k = 1/n_k$. The discrete setting uses cell centres (where ρ and ψ live) and grid vertices (where we parametrise the map T): since $T(x) \approx x + \nabla\psi(x)$, we build a *continuous* approximate map on a vertex backbone, so that adjacent cells share vertices and the resulting map is continuous. The scheme is thus cell-centred for densities but vertex-based for the map.

Cell-centre multi-indices are

$$i = (i_1, \ldots, i_d) \in \mathcal{I} := \prod_{k=1}^{d} \{0, \ldots, n_k - 1\},$$

with physical coordinates and cells

$$x_i := \left(\frac{i_1 + 1/2}{n_1}, \ldots, \frac{i_d + 1/2}{n_d} \right), \qquad C_i = \prod_{k=1}^{d} \left[\frac{i_k}{n_k}, \frac{i_k + 1}{n_k} \right],$$

and $\rho_i \approx \rho(x_i)$, $\psi_i \approx \psi(x_i)$. The auxiliary vertex grid is the Cartesian mesh of cell corners

$$\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathcal{V} := \prod_{k=1}^{d} \{0, \ldots, n_k\}, \qquad z_\alpha := \left(\frac{\alpha_1}{n_1}, \ldots, \frac{\alpha_d}{n_d} \right) \in [0,1]^d.$$

For a fixed cell i its 2^d vertices are $\alpha^{(m)}(i) = i + c^{(m)}$, $c^{(m)} \in \{0,1\}^d$, $m = 1, \ldots, 2^d$.

Discrete map T construction. Define the multilinear interpolant $I\psi : [0,1]^d \to \mathbb{R}$ of the cell-centred values ψ_i by

$$I\psi(x) = \sum_{c \in \{0,1\}^d} w_c(f)\ \psi_{b+c}, \qquad w_c(f) := \prod_{k=1}^{d} \big[c_k f_k + (1 - c_k)(1 - f_k) \big],$$

$$X_k = n_k x_k - 1/2, \qquad b_k = \lfloor X_k \rfloor, \qquad f_k = X_k - b_k \in [0,1).$$

At each vertex z_α we approximate $\nabla\psi$ by central differences of $I\psi$ (well defined at the vertex even though $I\psi$ is only C^0),

$$(\nabla_h\psi)(z_\alpha) := \big(\partial_1^h\psi(z_\alpha), \dots, \partial_d^h\psi(z_\alpha)\big), \qquad \partial_k^h\psi(z_\alpha) := \frac{I\psi(z_\alpha + h_k e_k) - I\psi(z_\alpha - h_k e_k)}{2h_k},$$

and set the vertex transport map

$$T_v(z_\alpha) := \mathrm{Proj}_{[0,1]^d}\big[z_\alpha + (\nabla_h\psi)(z_\alpha)\big],$$

where $\mathrm{Proj}_{[0,1]^d}$ is the componentwise clip onto $[0,1]$.

The pushforward, however, must act on the mass inside cells, $\rho'(y)\,\mathrm{d}y \approx T_\#(\rho(x)\,\mathrm{d}x)$. For each cell i we therefore define a cellwise map T_i by multilinear interpolation of its vertex images. Write a point $x \in C_i$ in local coordinates

$$x = x_i + \sum_{k=1}^{d} \big(t_k - \tfrac{1}{2}\big)\, h_k e_k, \qquad t = (t_1, \dots, t_d) \in [0,1]^d,$$

so $t_k = 0$ is the "left" side of the cell and $t_k = 1$ the "right". Then

$$T_i(x) := \sum_{m=1}^{2^d} \lambda_m(t)\, T_v\big(z_{\alpha^{(m)}(i)}\big), \qquad \lambda_m(t) := \prod_{k=1}^{d} \Big(c_k^{(m)} t_k + (1 - c_k^{(m)})(1 - t_k)\Big),$$

the same multilinear weights as w_c above, associated with the corners $c^{(m)}$. Assembling all cells gives a piecewise multilinear, globally continuous map

$$T : [0,1]^d \to [0,1]^d, \quad T(x) := T_i(x) \text{ if } x \in C_i, \qquad T(x) \approx x + \nabla\psi(x).$$

Stretch-based adaptive sampling per cell. For each cell i and axis k we estimate a *local stretch* of the map from the vertex images. Consider the edges of C_i parallel to axis k, i.e. vertex pairs $\alpha = i + c$ and $\alpha' = i + c + e_k$ with $c \in \{0,1\}^d$, $c_k = 0$ (so $\alpha_k = i_k$, $\alpha'_k = i_k + 1$), and set

$$s_{i,k} := \max_{c \in \{0,1\}^d,\, c_k = 0} \Big| T_v^{(k)}(z_{i+c+e_k}) - T_v^{(k)}(z_{i+c}) \Big|,$$

where $T_v^{(k)}$ is the k-th component of T_v. A large $s_{i,k}$ signals a large displacement of the k-th coordinate across the edge (a compression corresponds to a small $s_{i,k}$). Since $s_{i,k}/h_k = s_{i,k} n_k$ is the number of target cells the stretched edge spans, we take

$$M_{i,k} := \max\big(1, \lceil s_{i,k}\, n_k \rceil\big), \qquad S_i := \prod_{k=1}^{d} M_{i,k}$$

samples along axis k, so that each spanned target cell receives at least one sample. We place them at the sub-interval midpoints: for $l = (l_1, \dots, l_d)$ with $0 \le l_k \le M_{i,k} - 1$,

$$t_{i,l,k} := \frac{l_k + 1/2}{M_{i,k}}, \qquad t_{i,l} := (t_{i,l,1}, \dots, t_{i,l,d}) \in (0,1)^d, \qquad x_{i,l} = x_i + \sum_{k=1}^{d} \left(t_{i,l,k} - \tfrac{1}{2}\right) h_k e_k.$$

The mapped point is

$$y_{i,l} := T_i(x_{i,l}) = \sum_{m=1}^{2^d} \lambda_m(t_{i,l})\, T_v\big(z_{\alpha^{(m)}(i)}\big) \in [0,1]^d,$$

and each cell i of mass ρ_i is represented by S_i samples $x_{i,l}$, each carrying mass $m_{i,l} := \rho_i / S_i$.

Building the resulting density ρ'. We deposit the mapped samples back onto the cell-centre grid $\{x_j\}_{j \in \mathcal{I}}$ by the CIC assignment (Deserno and Holm, 1998; Hockney and Eastwood, 1988). For a mapped point $y \in [0,1]^d$ set

$$X_k := n_k y_k - 1/2, \qquad b_k := \lfloor X_k \rfloor, \qquad \delta_k := X_k - b_k \in [0,1),$$

clamped so that $b_k \in \{0, \dots, n_k - 2\}$; the two candidate indices along axis k are $j_k^{(0)} := b_k$ and $j_k^{(1)} := b_k + 1$, with weights

$$w_k^{(0)}(y) := 1 - \delta_k, \qquad w_k^{(1)}(y) := \delta_k.$$

For $j = (j_1, \dots, j_d)$ with each $j_k \in \{j_k^{(0)}, j_k^{(1)}\}$,

$$w_j(y) := \prod_{k=1}^{d} w_k^{(\sigma_k)}(y), \qquad \sigma_k = 0 \text{ if } j_k = j_k^{(0)}, \quad \sigma_k = 1 \text{ if } j_k = j_k^{(1)},$$

so each y deposits to at most 2^d centres with $\sum_j w_j(y) = 1$. A sample $y_{i,l}$ of mass $m_{i,l}$ contributes $\Delta\rho'_j = m_{i,l}\, w_j(y_{i,l})$, and summing over all cells and samples gives the un-normalised density

$$\tilde{\rho}'_j := \sum_{i \in \mathcal{I}} \sum_{l\,:\, 0 \le l_k \le M_{i,k}-1} \frac{\rho_i}{S_i}\, w_j(y_{i,l}), \qquad j \in \mathcal{I}.$$

Because of clipping and out-of-range deposits the total mass may drift, so we renormalise:

$$\rho'_j := \begin{cases} \tilde{\rho}'_j \cdot \dfrac{\overline{\rho}}{\overline{\tilde{\rho}'}} & \overline{\tilde{\rho}'} > 0, \\ \tilde{\rho}'_j & \text{otherwise,} \end{cases} \qquad \overline{\rho} := \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \rho_i, \quad \overline{\tilde{\rho}'} := \frac{1}{|\mathcal{I}|} \sum_{j \in \mathcal{I}} \tilde{\rho}'_j.$$

B.4 Convolutional Sinkhorn

The dense kernel applications Kv and $K^\top u$ (4.2.5) are the only expensive step of a Sinkhorn iteration, and on a regular grid they can be made almost free. With the squared-Euclidean cost

$C_{ij} = \frac{1}{2}\|x_i - x_j\|^2$ the Gibbs kernel

$$K_{ij} = \exp\Big(-\frac{C_{ij}}{\varepsilon}\Big) = \exp\Big(-\frac{\|x_i - x_j\|^2}{2\varepsilon}\Big) \tag{B.1}$$

is, up to a multiplicative constant, the *heat kernel* $H_t(x, y) = (4\pi t)^{-d/2} \exp\big(-\|x-y\|^2/4t\big)$ evaluated at diffusion time $t = \varepsilon/2$. The constant is irrelevant, as it is absorbed by the Sinkhorn scalings.

Since the squared distance is a sum over coordinates, $\|x - y\|^2 = \sum_{\ell=1}^{d} \big(x^{(\ell)} - y^{(\ell)}\big)^2$, the exponential factorises into a product of one-dimensional kernels,

$$K_{ij} = \prod_{\ell=1}^{d} \exp\Big(-\frac{(x_i^{(\ell)} - x_j^{(\ell)})^2}{2\varepsilon}\Big) = \big(K^{(1)} \otimes \cdots \otimes K^{(d)}\big)_{ij}, \tag{B.2}$$

so K is a tensor (Kronecker) product, and applying it to a density stored as a d-dimensional array reduces to one Gaussian convolution along each axis in turn. Along each axis this is a discrete convolution with a Gaussian stencil whose width is set by ε, so the dense kernel application $K(\cdot)$ is replaced by d cheap one-dimensional passes. We write $\mathrm{conv}_\varepsilon(\cdot) \equiv K(\cdot)$ for this operator, summarised in Algorithm 7.

Algorithm 6 Convolutional Sinkhorn barycenter

Input: marginals $\{\mu_j\}_{j=1}^{J}$, weights $\{\alpha_j\}$ with $\sum_j \alpha_j = 1$, regularization ε

$v_j \leftarrow \mathbf{1}$ for all j
repeat
  for $j = 1, \ldots, J$ **do**
    $u_j \leftarrow \mu_j \oslash \mathrm{conv}_\varepsilon(v_j)$ ▷ enforce the μ_j marginal
    $s_j \leftarrow \mathrm{conv}_\varepsilon(u_j)$
  end for
  $\nu \leftarrow \exp\Big(\sum_j \alpha_j \log\big(v_j \odot s_j\big)\Big)$ ▷ weighted geometric mean
  for $j = 1, \ldots, J$ **do**
    $v_j \leftarrow \nu \oslash s_j$ ▷ enforce the shared marginal ν
  end for
until converged
return ν

Algorithm 7 $\mathrm{conv}_\varepsilon$: separable Gaussian kernel application

Input: array a on a d-dimensional grid with spacing h, regularization ε

build the one-dimensional Gaussian stencil $g_m = \exp\big(-m^2h^2/2\varepsilon\big)$, normalised
for $\ell = 1, \ldots, d$ **do**
  $a \leftarrow$ convolve a with g along axis ℓ ▷ 1D pass
end for
return a

B.5 Back-and-Forth barycenter

We compute the barycenter by a damped fixed-point iteration in measure space, using the back-and-forth solver (Jacobs and Léger, 2020) as the inner Monge solver. For the quadratic cost, the

barycenter $\nu^\star$ of $(\mu_j)_{j=1}^J$ with weights (λ_j) is characterised (Agueh and Carlier, 2011) by the optimal Brenier maps $T_j : \nu^\star \to \mu_j$ satisfying the first-order condition

$$\sum_{j=1}^{J} \lambda_j \, T_j \;=\; \mathrm{id} \qquad \nu^\star\text{-a.e.} \tag{B.3}$$

This motivates the iteration of (Álvarez Esteban et al., 2016): given the current iterate ν_k, solve $\mathrm{OT}(\nu_k, \mu_j)$ for every j and push ν_k through the λ-averaged map. For the quadratic cost the dual is attained, so the back-and-forth solver returns the maximising Kantorovich potential ψ_j of the dual problem $\mathcal{D}(\nu_k, \mu_j)$, and by 2.3.6 the associated Brenier map is $T_j = \mathrm{id} - \nabla\psi_j$. Linearity of the gradient then makes the λ-averaged map *itself* a gradient map,

$$\bar{T}_k \;=\; \sum_{j=1}^{J} \lambda_j \, T_j \;=\; \mathrm{id} - \nabla\Psi_k, \qquad \Psi_k \;:=\; \sum_{j=1}^{J} \lambda_j \, \psi_j. \tag{B.4}$$

This is the one observation that makes the scheme cheap: the J maps are never averaged pointwise; it suffices to average their potentials and differentiate once. The barycenter is then updated by

$$\nu_{k+1} \;=\; (1-\tau)\,\nu_k \;+\; \tau\,(\bar{T}_k)_{\#}\nu_k, \qquad \tau \in (0,1], \tag{B.5}$$

with τ a relaxation parameter ($\tau = 1$ recovers the plain pushforward iteration). At a fixed point $\nu_\infty = (\bar{T}_\infty)_{\#}\nu_\infty$ one has $\bar{T}_\infty = \mathrm{id}$, which is exactly the optimality condition (B.3), so the limit is the desired barycenter. The procedure is well posed because each inner solve returns an *unregularized* Brenier map — the back-and-forth step introduces no entropic blur in the transport — while the relaxation τ damps the map averaging and stabilises the early iterations, when ν_k is still far from $\nu^\star$.

Algorithm 8 Back-and-Forth barycenter

Input: marginals $\{\mu_j\}_{j=1}^J$; weights $\{\lambda_j\}$ with $\sum_j \lambda_j = 1$; relaxation $\tau \in (0,1]$; iterations M; back-and-forth budget K, step sizes $\{\sigma_k\}$
Initialize: $\nu_0 \leftarrow \sum_j \lambda_j \, \mu_j$ ▷ or any density on Ω
for $k = 0, 1, \ldots, M-1$ **do**
  for $j = 1, \ldots, J$ **do**
    $\varphi_j,\, \psi_j \leftarrow \textsc{BackAndForth}(\nu_k, \mu_j; \{\sigma_k\}, K)$ ▷ Algorithm 2
  end for
  $\Psi_k \leftarrow \sum_{j=1}^J \lambda_j \, \psi_j$ ▷ average the potentials
  $\bar{T}_k \leftarrow \mathrm{id} - \nabla\Psi_k$ ▷ λ-averaged map, cf. (B.4)
  $\nu_{k+1} \leftarrow (1-\tau)\,\nu_k + \tau\,(\bar{T}_k)_{\#}\nu_k$ ▷ damped update (B.5)
end for
return ν_M

C Figures

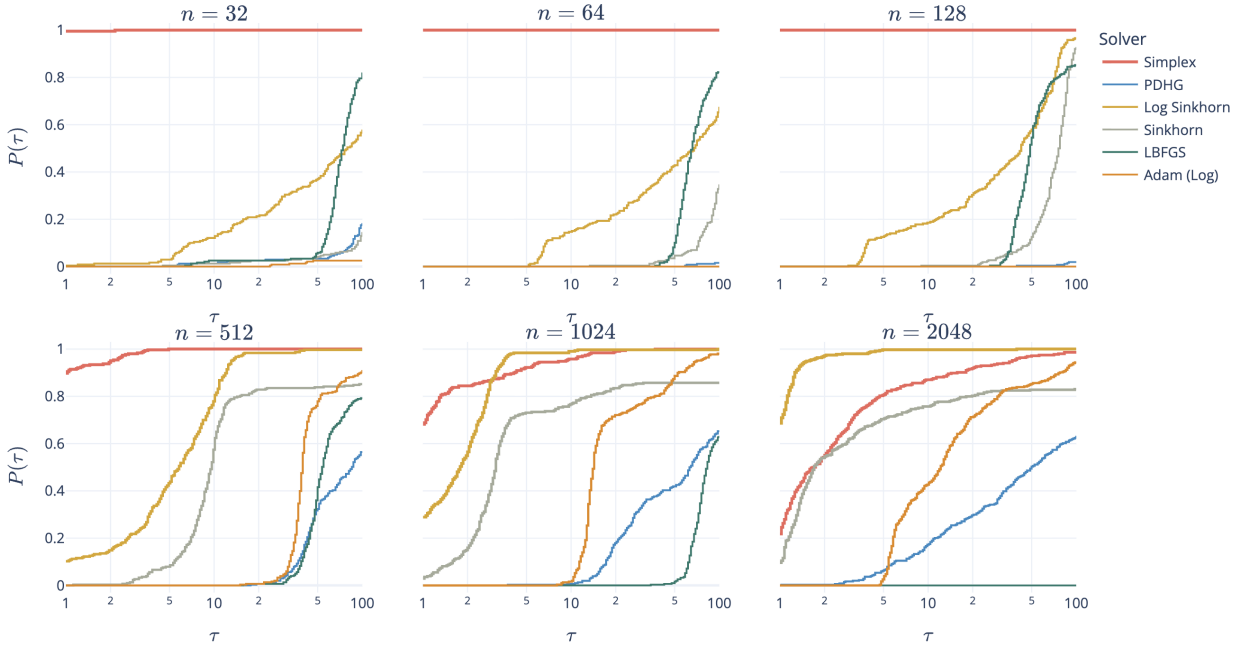
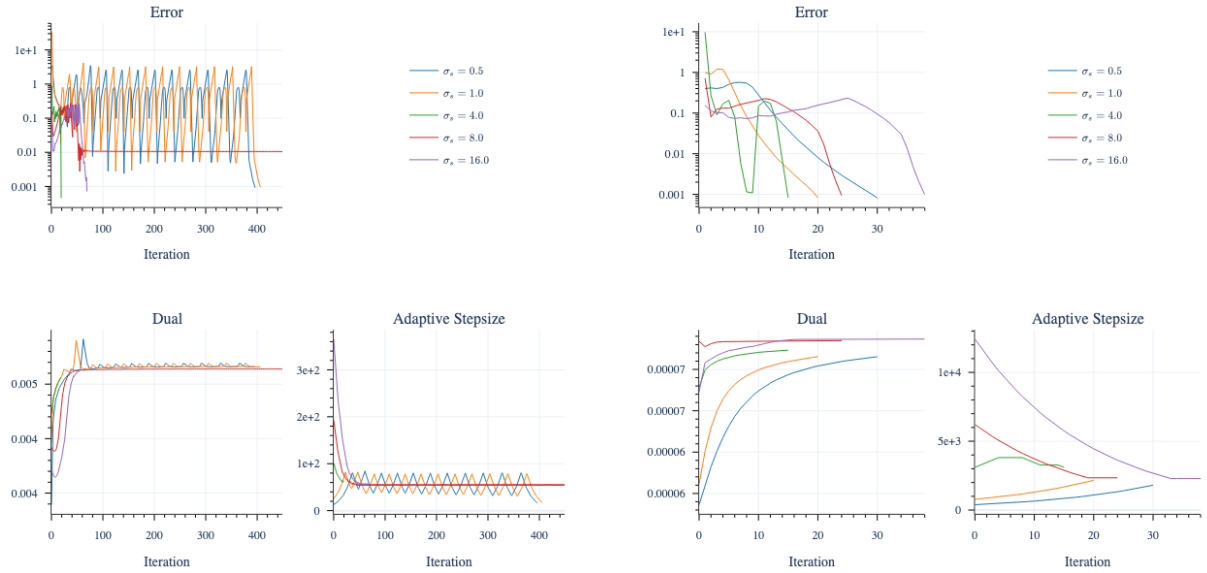


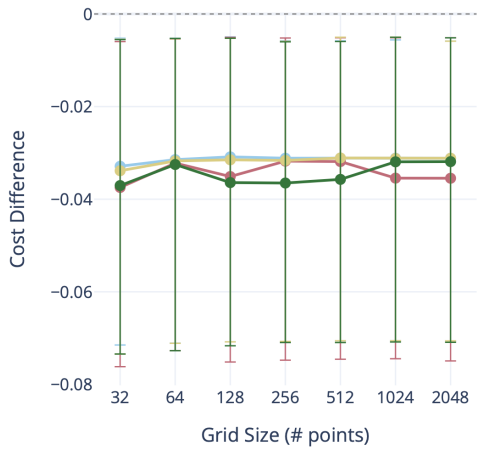
FIGURE 30. *Performance profiles by grid size* $n \in \{32, 64, 128, 512, 1024, 2048\}$ *(*$n = 256$ *is omitter to keep the clarity of the plot). Left intercept* $P_s(1)$ *indicates wins; the right plateau shows robustness.*



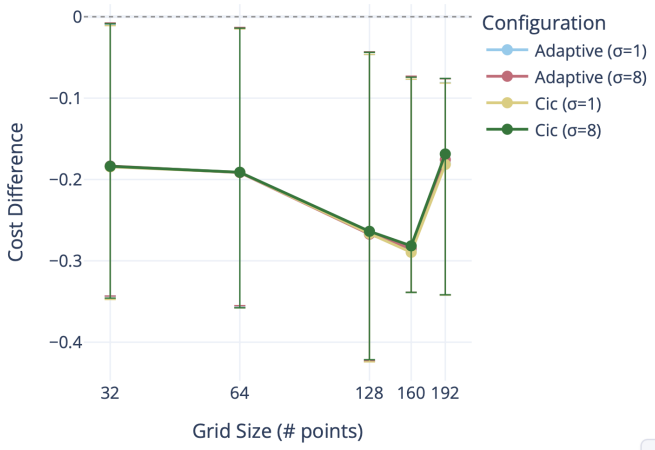
(A) *Convergence of the Back-and-Forth method on the 1D Gaussian benchmark for different initial Armijo–Goldstein stepsizes* $\sigma_s \in \{0.5, 1, 4, 8, 16\}$. *Top: decay of the relative* $\dot{H}^1$ *error over iterations. Bottom left: evolution of the dual objective. Bottom right: adaptive stepsize selected by the Armijo–Goldstein rule.*

(B) *Convergence of the Back-and-Forth method on the 2D Gaussian mixture benchmark for different initial stepsizes* $\sigma_s \in \{0.5, 1, 4, 8, 16\}$. *Top: decay of the relative* $\dot{H}^1$ *error. Bottom left: dual objective. Bottom right: adaptive stepsize chosen by the Armijo–Goldstein rule. In contrast to the 1D case, the error decreases regularly and converges within about* 30–35 *iterations for all tested stepsizes, while the line search damps overly aggressive initial steps.*

FIGURE 31. *Convergence plots of the Back-and-Forth method for various initial stepsize scales* $\sigma_s \in \{0.5, 1, 4, 8, 16\}$.



(A) *1D problems. For all grid sizes and configurations the Back-and-Forth cost lies consistently below the simplex benchmark, with a median gap of about* 3–4×10^{-2} *and wide but roughly symmetric variability around this level. There is no clear trend in the bias with respect to grid size or configuration.*



(B) *2D problems. The same systematic underestimation is present but more pronounced: median gaps are around* 2×10^{-1} *across all grid sizes. All four configurations behave almost identically, and the spread across instances is substantially larger than the differences between them.*

FIGURE 32. *Transport cost difference between the Back-and-Forth method and the simplex benchmark as a function of grid size in 1D and 2D. We plot the median and interquartile range of the quantity over all distributions; the dashed horizontal line corresponds to the simplex cost (*0 *difference). Negative values indicate that Back-and-Forth underestimates the optimal transport cost.*

D Tables

TABLE 9

Max-iteration hit rate (%) by distribution family for 1D experiments of the Back-and-Forth method. Rates are computed among runs that completed within the wall-clock budget.

Distribution	**Adaptive** $\sigma_s = 1$	**Adaptive** $\sigma_s = 8$	**CIC** $\sigma_s = 1$	**CIC** $\sigma_s = 8$
Gen. Hyperb. Mixt. (6 comp.)	15.7	34.8	21.0	33.8
Gen. Hyperb. Mixt. (4 comp.)	17.6	47.1	21.4	30.5
Gen. Hyperb. Mixt. (2 comp.)	20.0	46.7	30.5	38.1
Gaussian (6 comp.)	10.5	30.0	14.3	19.5
Gaussian (4 comp.)	12.9	36.2	22.9	28.6
Gaussian (2 comp.)	20.5	44.8	26.2	36.2
Gaussian	7.1	12.9	26.7	30.0
Exponential → Gaussian	13.4	42.9	12.4	18.0
Exponential → Cauchy	5.5	30.4	3.7	10.6
Exponential	12.9	40.0	23.8	32.9
Cauchy → Gaussian	20.7	85.7	21.7	35.5

TABLE 10

Max-iteration hit rate (%) of the Back-and-Forth method by distribution family for 2D experiments.

Distribution	**Adaptive** $\sigma_s = 1$	**Adaptive** $\sigma_s = 8$	**CIC** $\sigma_s = 1$	**CIC** $\sigma_s = 8$
Students (6 d.o.f.)	0.0	1.3	0.7	0.7
Students (4 d.o.f.)	0.0	1.3	0.7	1.3
Students (2 d.o.f.)	0.0	0.7	0.7	1.3
Gaussian (6 comp.)	0.0	0.0	0.7	1.3
Gaussian (4 comp.)	0.0	1.3	0.0	2.7
Gaussian (2 comp.)	0.7	3.3	0.0	2.7
Gaussian	0.0	1.7	0.0	0.0
Exponential → Gaussian	0.0	1.3	0.0	0.6
Exponential → Cauchy	0.0	5.2	0.0	9.7
Exponential	0.0	0.7	0.7	1.3
Cauchy → Gaussian	0.6	1.3	0.6	1.3
Cauchy	3.3	6.0	0.7	8.0

TABLE 11
Max-iteration hit rate (%) of the Back-and-Forth method by distribution family for 3D experiments.

Distribution	**Adaptive** $\sigma_s = 1$	**Adaptive** $\sigma_s = 8$	**CIC** $\sigma_s = 1$	**CIC** $\sigma_s = 8$
Students (6 d.o.f.)	0.0	4.4	0.0	1.1
Students (4 d.o.f.)	0.0	4.4	0.0	1.1
Students (2 d.o.f.)	0.0	4.4	0.0	1.1
Gaussian (6 comp.)	0.0	0.0	0.0	0.0
Gaussian (4 comp.)	0.0	0.0	0.0	0.0
Gaussian (2 comp.)	0.0	0.0	0.0	0.0
Gaussian	0.0	0.0	0.0	0.0
Exponential	2.2	2.2	0.0	0.0
Cauchy	2.2	13.3	0.0	17.8

TABLE 12
Aggregated marginal error between the numerical pushforward and the target distribution for the Back-and-Forth method, pooled over all grid sizes and distributions. Entries report the median and interquartile range (IQR) over converged runs for each dimension and configuration.

Dim.	**Scheme**	σ_s	**Median error**	**IQR**
1D	CIC	1	0.017	[0.0074, 0.038]
	CIC	8	0.023	[0.0109, 0.0484]
	Adaptive	1	0.0053	[0.0024, 0.0122]
	Adaptive	8	0.014	[0.0068, 0.0295]
2D	CIC	1	0.0051	[0.0029, 0.0117]
	CIC	8	0.0084	[0.0044, 0.0167]
	Adaptive	1	0.0020	[0.0011, 0.0037]
	Adaptive	8	0.0040	[0.0016, 0.0105]
3D	CIC	1	0.0125	[0.0058, 0.0242]
	CIC	8	0.0142	[0.0065, 0.0283]
	Adaptive	1	0.0050	[0.0011, 0.0107]
	Adaptive	8	0.0097	[0.0017, 0.0203]

TABLE 13

Monotonicity fraction of the Back-and-Forth maps, pooled over all grid sizes and distributions. The monotonicity fraction is the proportion of grid points where the symmetric part of the discrete Jacobian ∇T has a non-negative smallest eigenvalue, up to numerical tolerance; equivalently, where T is locally monotone. Since a Brenier map satisfies $T = \nabla\varphi$ for convex φ, this is the fraction where the implied Hessian $\nabla^2\varphi = \nabla T$ is positive semidefinite, so values close to 1 indicate preservation of the Brenier structure. Entries report the median and IQR over all converged runs.

Dim.	**Scheme**	σ_s	**Median**	**IQR**
1D	CIC	1	1.0000	[1, 1]
	CIC	8	1.0000	[1, 1]
	Adaptive	1	1.0000	[1, 1]
	Adaptive	8	1.0000	[1, 1]
2D	CIC	1	0.9998	[0.9956, 1]
	CIC	8	0.9971	[0.9854, 1]
	Adaptive	1	0.9999	[0.9951, 1]
	Adaptive	8	0.9983	[0.9886, 1]
3D	CIC	1	0.9982	[0.9817, 1]
	CIC	8	0.9928	[0.9590, 1]
	Adaptive	1	0.9993	[0.9729, 1]
	Adaptive	8	0.9988	[0.9580, 1]

References

Ivan Zhytkevych, Denys Ruban, and Augusto Gerolin. Back-and-forth flows for optimization problems in the space of probability measures. Manuscript under review, 2026.

Filippo Santambrogio. *Optimal Transport for Applied Mathematicians*, volume 87 of *Progress in Nonlinear Differential Equations and Their Applications*. Birkhäuser, Cham, 2015. doi: 10.1007/978-3-319-20828-2.

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019. doi: 10.1561/2200000073.

Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013. URL `https://arxiv.org/abs/1306.0895`.

Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional wasserstein distances. *ACM Transactions on Graphics*, 34(4):66:1–66:11, 2015. doi: 10.1145/2766963. URL `https://hal.science/hal-01188953`.

Victor M. Panaretos and Yoav Zemel. *An Invitation to Statistics in Wasserstein Space*. SpringerBriefs in Probability and Mathematical Statistics. Springer, Cham, 1 edition, 2020. ISBN 978-3-030-38438-8. doi: 10.1007/978-3-030-38438-8. URL `https://doi.org/10.1007/978-3-030-38438-8`. xiii + 147 pages.

Alfred Galichon. *Optimal Transport Methods in Economics*. Princeton University Press, 2016. URL `http://www.jstor.org/stable/j.ctt1q1xs9h`.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/arjovsky17a.html`.

Matt Jacobs and Flavien Léger. A fast approach to optimal transport: the back-and-forth method. *Numerische Mathematik*, 146(3):513–544, 2020. ISSN 0945-3245. doi: 10.1007/s00211-020-01154-8. URL `https://doi.org/10.1007/s00211-020-01154-8`.

Denys Ruban and Augusto Gerolin. Primal-dual hybrid algorithms for chi-squared regularized optimal transport: statistical-computational trade-offs and applications to wasserstein barycenters. In *OPT 2025: 17th Annual Workshop on Optimization for Machine Learning at NeurIPS 2025*, 2025. URL `https://opt-ml.org/papers/2025/paper45.pdf`. Workshop paper.

Jörn Schrieber, Dominic Schuhmacher, and Carsten Gottschlich. Dotmark – a benchmark for discrete optimal transport. *IEEE Access*, 5:271–282, 2017. doi: 10.1109/ACCESS.2016.2639065. URL `https://arxiv.org/abs/1610.03368`.

Yihe Dong, Yu Gao, Richard Peng, Ilya Razenshteyn, and Saurabh Sawlani. A study of performance of optimal transport, 2020. URL `https://arxiv.org/abs/2005.01182`.

Alexander Korotin, Lingxiao Li, Aude Genevay, Justin Solomon, Alexander Filippov, and Evgeny Burnaev. Do neural optimal transport solvers work? A continuous Wasserstein-2 benchmark. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL `https://arxiv.org/abs/2106.01954`.

Quentin Mérigot and Boris Thibert. Optimal transport: discretization and algorithms. In *Geometric Partial Differential Equations – Part II*, Handbook of Numerical Analysis, pages 133–212. Elsevier, 2021. doi: 10.1016/bs.hna.2020.10.001. URL https://arxiv.org/abs/2003.00855.

Sina Moradi. A survey on algorithmic developments in optimal transport problem with applications, 2025. URL https://arxiv.org/abs/2501.06247.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T. H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://arxiv.org/abs/1605.08527.

Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958. doi: 10.2140/pjm.1958.8.171.

Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on Pure and Applied Mathematics*, 44(4):375–417, 1991. doi: 10.1002/cpa.3160440402. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160440402.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89 of *Proceedings of Machine Learning Research*, pages 2681–2690. PMLR, 2019.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Marco Cuturi and Gabriel Peyré. A smoothed dual approach for variational Wasserstein problems. *SIAM Journal on Imaging Sciences*, 9(1):320–343, 2016. doi: 10.1137/15M1032600.

Simone Di Marino and Augusto Gerolin. An optimal transport approach for the Schrödinger bridge problem and convergence of Sinkhorn algorithm. *Journal of Scientific Computing*, 85(2):27, 2020. ISSN 1573-7691. doi: 10.1007/s10915-020-01325-7. URL https://doi.org/10.1007/s10915-020-01325-7. Article 27.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.6980.

Haim Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Universitext. Springer, New York, 2011.

David G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, New York, 1969.

Yves Lucet. Faster than the fast legendre transform, the linear-time legendre transform. *Numerical Algorithms*, 16(2):171–185, 1997. ISSN 1572-9265. doi: 10.1023/A:1019191114493. URL https://doi.org/10.1023/A:1019191114493.

Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966.

A. A. Goldstein. On steepest descent. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 3(1):147–151, 1965.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Number 87 in Applied Optimization. Springer New York, New York, NY, 2004. ISBN 978-1-4419-8853-9. doi: 10.1007/978-1-4419-8853-9. URL https://link.springer.com/book/10.1007/978-1-4419-8853-9.

Alexander J. McNeil, Rüdiger Frey, and Paul Embrechts. *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton Series in Finance. Princeton University Press, Princeton, NJ, 2005. ISBN 978-0-691-12255-7.

Karsten Prause. *The generalized hyperbolic model: Estimation, financial derivatives, and risk measures*. PhD thesis, University of Freiburg, 1999. URL `https://freidok.uni-freiburg.de/data/15`.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17:261–272, 2020.

Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Number 55 in National Bureau of Standards Applied Mathematics Series. U.S. Government Printing Office, Washington, D.C., 1964.

Ole E. Barndorff-Nielsen and Christian Halgreen. Infinite divisibility of the hyperbolic and generalized inverse Gaussian distributions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 38(4):309–311, 1977. doi: 10.1007/BF00533162.

Ryan P. Browne and Paul D. McNicholas. A mixture of generalized hyperbolic distributions. *The Canadian Journal of Statistics*, 43(2):176–198, 2015. doi: 10.1002/cjs.11246.

Antonin Chambolle and Juan Pablo Contreras. Accelerated Bregman primal-dual methods applied to optimal transport and Wasserstein barycenter problems. *SIAM Journal on Mathematics of Data Science*, 4(4):1369–1395, 2022. doi: 10.1137/22M1481865.

Bernhard Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, 2019. doi: 10.1137/16M1106018.

Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002. doi: 10.1007/s101070100263.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, Heidelberg, 2009. ISBN 978-3-540-71050-9. doi: 10.1007/978-3-540-71050-9.

Asuka Takatsu. Wasserstein geometry of gaussian measures. *Osaka Journal of Mathematics*, 48(4):1005–1026, 2011. URL `https://projecteuclid.org/journals/osaka-journal-of-mathematics/volume-48/issue-4/Wasserstein-geometry-of-Gaussian-measures/ojm/1326291215.full`.

Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011. doi: 10.1137/100805741. URL `https://doi.org/10.1137/100805741`.

Lingxiao Li, Aude Genevay, Mikhail Yurochkin, and Justin Solomon. Continuous regularized Wasserstein barycenters. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/cdf1035c34ec380218a8cc9a43d438f9-Abstract.html`.

Lénaïc Chizat. Doubly regularized entropic wasserstein barycenters, 2025. URL `https://arxiv.org/abs/2303.11844v2`. Version 2.

H. Sheikh Faridul, T. Pouli, C. Chamaret, J. Stauder, E. Reinhard, D. Kuzovkin, and A. Tremeau. Colour mapping: A review of recent methods, extensions and applications. *Computer Graphics Forum*, 35(1):59–88, 2016. doi: 10.1111/cgf.12671. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12671`.

Francois Pitié. Advances in colour transfer. *IET Computer Vision*, 14(6):304–322, 2020. doi: 10.1049/iet-cvi.2019.0920. URL `https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-cvi.2019.0920`.

Oriel Frigo, Neus Sabater, Vincent Demoulin, and Pierre Hellier. Optimal transportation for example-guided color transfer. In *Computer Vision – ACCV 2014*, volume 9007 of *Lecture Notes in Computer Science*, pages 655–670. Springer, 2015. doi: 10.1007/978-3-319-16811-1_43.

Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 97–104, 2011. doi: 10.1109/CVPR.2011.5995332.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018.

Michel X. Goemans. Linear programming. Lecture notes, 18.310A Principles of Discrete Applied Mathematics, Massachusetts Institute of Technology, March 2015. Dated March 17, 2015. `https://math.mit.edu/~goemans/18310S15/lpnotes310.pdf`.

Markus Deserno and Christian Holm. How to mesh up ewald sums. i. a theoretical and numerical comparison of various particle mesh routines. *The Journal of Chemical Physics*, 109(18):7678–7693, November 1998. ISSN 1089-7690. doi: 10.1063/1.477414. URL `http://dx.doi.org/10.1063/1.477414`.

R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. Adam Hilger, Bristol, 1988.

Pedro C. Álvarez Esteban, E. del Barrio, J. A. Cuesta-Albertos, and C. Matrán. A fixed-point approach to barycenters in Wasserstein space. *Journal of Mathematical Analysis and Applications*, 441(2):744–762, 2016. doi: 10.1016/j.jmaa.2016.04.045. URL `https://arxiv.org/abs/1511.05355`.