Optimal Transport for Probabilistic Circuits

Adrian Ciotinga[1] **YooJung Choi**[1]

[1]School of Computing and Augmented Intelligence, Arizona State University

Abstract

We introduce a novel optimal transport framework for probabilistic circuits (PCs). While it has been shown recently that divergences between distributions represented as certain classes of PCs can be computed tractably, to the best of our knowledge, there is no existing approach to compute the Wasserstein distance between probability distributions given by PCs. We propose a Wasserstein-type distance that restricts the coupling measure of the associated optimal transport problem to be a probabilistic circuit. We then develop an algorithm for computing this distance by solving a series of small linear programs and derive the circuit conditions under which this is tractable. Furthermore, we show that we can easily retrieve the optimal transport plan between the PCs from the solutions to these linear programs. Lastly, we study the empirical Wasserstein distance between a PC and a dataset, and show that we can estimate the PC parameters to minimize this distance through an efficient iterative algorithm.

1 INTRODUCTION

The Wasserstein distance is a statistical distance metric corresponding to the objective value taken by the optimal transport problem as proposed by Kantorovich's optimal transport framework that, given two probability measures, finds its optimal value at a coupling measure where the expected distance between the original two measures is minimized [Kantorovich, 1942]. Computing such a distance has proven extremely useful, with applications in generative modeling [Arjovsky et al., 2017], data privacy [Li et al., 2007], and distributionally robust optimization [Rahimian and Mehrotra, 2022]. Providing a detailed mapping between probability measures, optimal transport maps have also proven useful for problems such as fairness auditing [Black et al., 2020] and as generative models on their own [Rout et al., 2022a].

Unfortunately, computing the Wasserstein distance efficiently and exactly is a difficult task for all but the simplest distributions. This has led to the popularity of Wasserstein-like distances that enable tractable computation, such as the Sliced Wasserstein distance [Bonneel et al., 2015], Tree Wasserstein distance [Le et al., 2019] and the Mixture Wasserstein distance Delon and Desolneux [2020]. Others address this challenge by finding approximate solutions, with common approaches being the Sinkhorn Cuturi [2013] and factored couplings [Forrow et al., 2019] optimal transport solvers.

Modeling probability distributions in a way that enables tractable computation of queries is of great interest to the machine learning community. Probabilistic circuits (PCs) [Choi et al., 2020] provide a unifying framework for representing many classes of tractable probabilistic models as computational graphs; within this framework, tractability of certain queries can be guaranteed by imposing structural properties on the computational graph of the circuit. This includes tractable marginal and conditional inference, as well as pairwise queries that compare two distributions such as Kullback-Leibler (KL) divergence and cross-entropy [Liang and Van den Broeck, 2017, Vergari et al., 2021]. However, to the best of our knowledge, there is no existing algorithm that tractably computes the Wasserstein distance between two probabilistic circuits.

While algorithms for computing other statistical distance measures between PCs are well-established, the Wasserstein distance offers a distinct advantage in many applications. Measures such as KL divergence and cross-entropy are unbounded between distributions with disjoint supports; on the other hand, the p-Wasserstein distance is always bounded for distributions with finite p-th moments [Villani, 2008, p. 107]. Computing the Wasserstein distance also provides a bound for other statistical distance metrics such as the

Prokhorov metric and the total-variation distance [Gibbs and Su, 2002]. However, computing the Wasserstein distance is often intractable, even for models that are tractable for other probabilistic queries such as the KL divergence or cross-entropy, due to the inherent optimization problem required to be solved.

This paper focuses on computing (or bounding) the Wasserstein distance and optimal transport plan (i) between two PCs and (ii) between a PC and an empirical distribution. For (i) we propose a Wasserstein-type distance that upper-bounds the true Wasserstein distance and provide an efficient and exact algorithm to compute it between two circuits (Section 3). For (ii) we propose a parameter estimation algorithm for PCs that seeks to minimize the Wasserstein distance between a circuit and an empirical distribution (Section 4). We empirically evaluate our proposed methods on randomly generated PCs as well as on a benchmark dataset (Section 5).

2 PRELIMINARIES

We use capital letters (X) to denote random variables and lowercase letters (x) to denote their assignments. Boldface denotes a set of random variables and their assignments respectively (e.g., $\mathbf{X}$ and $\mathbf{x}$).

Wasserstein Distance and Optimal Transport Let P and Q be probability measures on metric space $\mathbb{R}^n$. For $p \geq 1$, the *p-Wasserstein distance* between P and Q is defined as:

$$\mathsf{W}_p^p(P, Q) = \inf_{\gamma \in \Gamma(P,Q)} \mathbb{E}_{\gamma(\mathbf{x},\mathbf{y})}[\|\mathbf{x} - \mathbf{y}\|_p^p] \quad (1)$$

where $\|\cdot\|_p$ denotes the vector p-norm and $\Gamma(P, Q)$ denotes the set of all *couplings*, i.e. joint distributions whose marginal distributions match P and Q. That is, the following holds for all $\gamma \in \Gamma(P, Q)$:

$$P(\mathbf{x}) = \int_{\mathbb{R}^n} \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y}, \quad Q(\mathbf{y}) = \int_{\mathbb{R}^n} \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x}. \quad (2)$$

Here, the *Wasserstein objective* of a (not necessarily optimal) coupling refers to the expectation inside the infimum in Equation 1 taken over that coupling, and the *Wasserstein distance* between two distributions refers to the value taken by the Wasserstein objective for the optimal coupling. It can be shown that there is always a coupling that obtains the infimum above [Villani, 2008]. Such optimal coupling γ^* induces a *transport plan* $\mathbf{x} \mapsto \gamma^*(\mathbf{x}, .)$. If the coupling is deterministic, this is called a *transport map* $\mathbf{x} \mapsto T(x)$ where $T(x)$ is the support of $\gamma^*(\mathbf{x}, .)$.

Probabilistic Circuits Many tractable probabilistic models—arithmetic circuits [Darwiche, 2003], sum-product networks [Poon and Domingos, 2011], cutset networks [Rahman et al., 2014], and more—can be understood through a unifying framework of *probabilistic circuits* [Choi et al., 2020].

Definition 1. *A probabilistic circuit (PC) C over a set of variables $\mathbf{X}$ is a parameterized, rooted directed acyclic graph (DAG) with three types of nodes: sum, product, and input nodes. Each input node n is associated with function f_n that encodes a probability distribution over a variable $X_i \in \mathbf{X}$, also called its* scope $\mathsf{sc}(n)$. *The set of child nodes for an internal node (sum or product) n is denoted $\mathsf{ch}(n)$, and its scope is given by $\mathsf{sc}(n) = \bigcup_{c \in \mathsf{ch}(n)} \mathsf{sc}(c)$. Each sum node n has normalized parameters $\theta_{n,c}$ for each child node c. For an assignment $\mathbf{x}$, let $\mathbf{x}_n$ denote its projection onto the scope $\mathsf{sc}(n)$ of node n. Then a PC rooted at node n recursively defines a probability distribution $p_n(\mathbf{x})$ as follows:*

$$p_n(\mathbf{x}) = \begin{cases} f_n(\mathbf{x}) & \text{if } n \text{ is an input node,} \\ \prod_{c \in \mathsf{ch}(n)} p_c(\mathbf{x}_c) & \text{if } n \text{ is a product node,} \\ \sum_{c \in \mathsf{ch}(n)} \theta_{n,c} p_c(\mathbf{x}_c) & \text{if } n \text{ is a sum node.} \end{cases}$$

Probabilistic circuits admit exact and efficient computation of many probabilistic inference queries, enabled by enforcing certain structural constraints. In particular, throughout this paper we assume two properties, *smoothness* and *decomposability*, which enable tractable (polytime) computation of marginal and conditional queries. A PC is *smooth* if every sum node $n \in C$ has the same scope as its children: $\forall n_i \in \mathsf{ch}(n), \mathsf{sc}(n_i) = \mathsf{sc}(n)$; it is *decomposable* if the children of every product node $n \in C$ have disjoint scopes: $\forall n_i \neq n_j \in \mathsf{ch}(n), \mathsf{sc}(n_i) \bigcap \mathsf{sc}(n_j) = \emptyset$. Tractable computation of different query classes can be enabled by ensuring additional properties.[1] Critically, enforcing these structural properties does not restrict the *expressivity* of PCs (i.e., they can still represent any distribution), but may decrease their *expressive efficiency* (i.e., an exponentially-sized circuit may be required when enforcing a constraint).

3 OPTIMAL TRANSPORT BETWEEN PROBABILISTIC CIRCUITS

We now consider the problem of computing Wasserstein distances and optimal transport plans between distributions represented by probabilistic circuits. For notational simplicity, suppose $P(\mathbf{X})$ and $Q(\mathbf{Y})$ are two PCs defining probability measures on a metric space, with a bijective mapping between variables in $\mathbf{X}$ and those in $\mathbf{Y}$; w.l.o.g., let X_i and Y_i map to each other. We also assume that the input distributions in the PCs are univariate,[2] and also allow constant-time computation of the Wasserstein distance—following the standard assumption of tractability of input distributions for tractable inference on PCs. In particular, this is the

[1] E.g., KL divergence between two *compatible* and *deterministic* PCs can be computed in polynomial time [Vergari et al., 2021].

[2] We assume univariate input nodes for notational simplicity, but this is not a requirement.

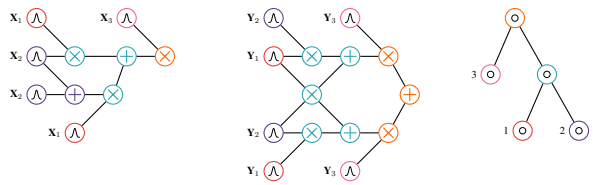

Figure 1: Compatible circuits over $\mathbf{X} = \{X_1, X_2, X_3\}$ and $\mathbf{Y} = \{Y_1, Y_2, Y_3\}$. Nodes in the same color have same scope, and the scope decomposition is visualized on the right.

case for p-Wasserstein distance between categorical distributions and for the 2-Wasserstein distance between Gaussian distributions. Unfortunately, even with these assumptions, computing the Wasserstein distance between probabilistic circuits is computationally hard.

Theorem 1. *Suppose P and Q are probabilistic circuits over n Boolean variables. Then computing the ∞-Wasserstein distance between P and Q is coNP-hard.*

In fact, the above is true even when the PCs satisfy stronger structural constraints (determinism and structured decomposability) that enable tractable inference of hard queries such as maximum-a-posteriori (MAP) Choi and Darwiche [2017] and entropy [Vergari et al., 2021] and even closed-form maximum-likelihood parameter estimation. At a high level, the proof proceeds by reducing from the problem of deciding consistency of two OBDDs (a type of deterministic and structured-decomposable circuits) which is NP-hard [Meinel and Theobald, 1998, Lemma 8.14]. In particular, given the two OBDDs, we can construct two deterministic and structured-decomposable PCs in polynomial time such that the input OBDDs are consistent iff W_∞ between the PCs is not 1. We refer to Appendix B.1 for a detailed proof.

Theorem 1 shows that computing the ∞-Wasserstein distance between two PCs is computationally hard. Whether computing W_p for some other fixed p (such as $p = 1$ or 2) is NP-hard is still an open question—there only exist efficient algorithms that bound this quantity between GMMs (which can be viewed as a special case of PCs), rather than compute it exactly [Delon and Desolneux, 2020, Chen et al., 2018]. However, we are interested in efficiently computing or upper-bounding W_p for *arbitrary* p, including W_∞. Thus, to address this computational challenge, we consider a Wasserstein-type distance between PCs by restricting the set of coupling measures to be PCs of a particular structure. Furthermore, we derive the structural conditions on the input PCs required to construct such structure and find the parameters that minimize the Wasserstein objective in time quadratic in the size of the input circuits.

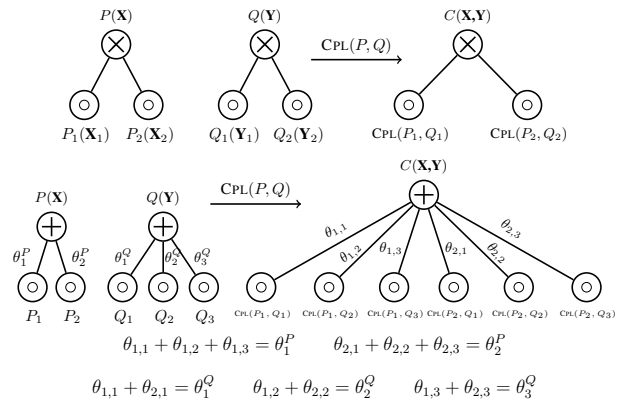

Figure 2: Recursive construction of coupling circuits. (Top) Product nodes couple children with corresponding scopes. (Bottom) Sum nodes couple the Cartesian product of children, with marginal constraints for the parameters.

3.1 CW_p: A DISTANCE BETWEEN CIRCUITS

The building blocks of PCs are sum, product, and input nodes. Sum nodes encode mixtures of their child distributions; algorithms to compute optimal transport distances between mixtures of distributions are well-studied [Delon and Desolneux, 2020, Chen et al., 2018], and formulate these distances as a solution to the weighted discrete optimal transport problem between mixture components in each distribution. Next, product nodes encode factorizations of distributions; since the p-norm $\|\cdot\|_p^p$ is *separable* across dimensions, W_p between two factorized distributions is the sum of the Wasserstein distances between the corresponding factors. Lastly, input nodes provide a natural base-case for an optimal transport framework: we assume that we can compute the Wasserstein distance and corresponding transport plan between two input nodes in constant-time, which is the usual assumption for tractable inference using PCs. Given this, we ask a natural question: can optimal transport distances and transport plans between PCs be computed recursively?

We show that the answer is yes. To do this, we propose the notion of *coupling circuits* between two *compatible* PCs, and introduce a Wasserstein-type distance CW_p which restricts the coupling set in Equation 1 to be circuits of this form. We then exploit the structural properties guaranteed by coupling circuits to derive efficient algorithms for computing CW_p and associated transport plan.

Definition 2 (Circuit compatibility [Vergari et al., 2021])**.** *Two smooth and decomposable PCs P and Q over RVs $\mathbf{X}$ and $\mathbf{Y}$, respectively, are* compatible *if the following two conditions hold: (i) there is a bijective mapping $\leftrightarrow$ between RVs X_i and Y_i, and (ii) any pair of product nodes $n \in P$ and $m \in Q$ with the same scope up to the bijective mapping are mutually compatible and decompose the scope the same way—that is, if n and m have scopes $\mathbf{X}$ and $\mathbf{Y}$ and $\mathbf{X} \leftrightarrow \mathbf{Y}$,*

then n and m have the same number of children, and for each child of n with scope $\mathbf{X}_i$ there is a corresponding child of m with scope $\mathbf{Y}_i$ such that $\mathbf{X}_i \leftrightarrow \mathbf{Y}_i$. Such pair of nodes are called corresponding *nodes.*

That is, two circuits are compatible if they have the same *hierarchical scope partitioning*; see Figure 1 for an example pair of compatible circuits. This does not require the circuit structures to be the same.

Definition 3 (Coupling circuit)**.** *A* coupling circuit *C between two compatible PCs P and Q with scopes $\mathbf{X}$ and $\mathbf{Y}$, respectively, is a PC with the following properties. (i) Each node $r \in C$ is recursively a coupling of a pair of nodes $n \in P$ and $m \in Q$.*[3] *(ii) Each node $r \in C$ that is a coupling of sum nodes $n \in P, m \in Q$ with edge weights $\{\theta_i\}$ and $\{\theta_j\}$ has edge weights $\{\theta_{i,j}\}$ such that $\sum_i \theta_{i,j} = \theta_j$ and $\sum_j \theta_{i,j} = \theta_i$ for all i and j.*

Note that the above definition does not assign parameters to a coupling circuit structure; rather, the properties of a coupling circuit restrict the set of feasible parameters. The second property described above ensures that coupling circuit C satisfies the marginal-matching constraints in Equation 2 with respect to P and Q (proof in Appendix B.2) by requiring that the sub-circuit rooted at any internal node in the coupling circuit matches marginals to the corresponding nodes in the original circuits (which is a stronger constraint than the entire coupling circuit simply matching marginal distributions to the original circuits). We are now ready to define our proposed distance metric between PCs, which is the minimum Wasserstein objective obtained by a valid parameterization of their coupling circuit.

Definition 4 (Circuit Wasserstein distance)**.** *Let $P(\mathbf{X})$ and $Q(\mathbf{Y})$ be compatible PCs and $C_\theta(\mathbf{X}, \mathbf{Y})$ a coupling circuit parameterized by θ. The p-*Circuit Wasserstein distance *between P and Q is defined as:*

$$\mathsf{CW}_p^p(P, Q) = \inf_\theta \mathbb{E}_{C_\theta(\mathbf{x},\mathbf{y})}[\|\mathbf{x} - \mathbf{y}\|_p^p].$$

We now investigate some properties of CW_p. First, we note that CW_p upper-bounds the true Wasserstein distance between PCs as both are infima of the same Wasserstein objective, while the feasible set of couplings for CW_p is more restrictive. Moreover, CW_p is a metric on any set of compatible circuits, which is contrary to some other statistical measures such as KL divergence used to compare distributions. See Appendix B.3 for a formal proof.

Proposition 1. *For any set $\mathcal{C}$ of compatible circuits, CW_p defines a metric on $\mathcal{C}$.*

[3]The coupling circuit has the same structure (but not parameters) as the product circuit [Vergari et al., 2021] of P and Q. Informally, we construct a *cross product* of children at each pair of sum nodes, and the product of corresponding children at each pair of product nodes.

Lastly, the optimal coupling circuit $C(\mathbf{X}, \mathbf{Y})$ corresponding to $\mathsf{CW}_p(P, Q)$ induces a (albeit not necessarily optimal) transport plan that maps a point $\mathbf{x}$ to a distribution $C(\mathbf{Y}|\mathbf{X} = \mathbf{x})$ and vice versa.

3.2 TRACTABLE COMPUTATION OF CW_p

The structure of coupling circuits enables efficient computation of both the Circuit Wasserstein distance and optimal coupling circuit parameters. Informally, the CW_p optimization problem at a coupling circuit's sum node is simply the weighted discrete optimal transport problem, where weights are the original sum node parameters and distances are the CW_p distances between child circuits. The CW_p optimization problem also decomposes at coupled product nodes, as product nodes encode factored distributions. Thus, we leverage the desirable properties of coupling circuits to exactly and efficiently computes the Circuit Wasserstein distance of two compatible PCs, as well as the optimal coupling circuit parameters. The key observation enabling our algorithm is that the Wasserstein objective for a given parameterization of the coupling circuit can be computed recursively through a single feedforward pass through the circuit, and can also be minimized over its parameters in a single forward pass.

Recursive Computation of the Wasserstein Objective Let $C(\mathbf{X}, \mathbf{Y})$ be a parameterized coupling circuit and $g(n) = \mathbb{E}_{C_n}[\|\mathbf{x} - \mathbf{y}\|_p^p]$ the corresponding CW_p-objective function at each node $n \in C$. We can write $g(n)$ recursively as follows (see Appendix B.4 for correctness proof):

$$g(n) = \begin{cases} \mathsf{W}_p^p(c_1, c_2) & \text{if } n \text{ is } \otimes \text{ of input nodes,} \\ \sum_{c \in \mathsf{ch}(n)} g(c) & \text{if } n \text{ is } \otimes \text{ of internal nodes,} \\ \sum_{c \in \mathsf{ch}(n)} \theta_{n,c} g(c) & \text{if } n \text{ is } \oplus \text{ node.} \end{cases} \tag{3}$$

Thus, we can push computation of the Wasserstein objective down to the leaf nodes of a coupling circuit, and our algorithm only requires a closed-form solution for W_p between univariate input distributions as the base case.

Recursive Computation of the Optimal Coupling Circuit Parameters for CW_p Leveraging the recursive properties of the Wasserstein objective, we can compute the optimal parameters of the coupling circuit by solving a small linear program at each sum node. Algorithm 1 details the construction of a coupling circuit (illustration in Figure 2) and finding the optimal parameters to compute CW_p.

Specifically, we wish to find $\min_\theta g(n)$ where $g(n)$ can be written recursively as in Equation 3. By this definition, at sum nodes we can minimize the Wasserstein objective at each child independently then solve a linear program using the objective value at the child nodes as constants: given the optimal $g(c)$ for each child node c of n, we can rewrite $\min_\theta g(n) = \min_\theta \sum_{c \in \mathsf{ch}(n)} \theta_c g(c)$ to see that solving for the sum node parameters reduces to

Algorithm 1 COUPLE(n, m): coupling circuit that optimizes $\mathsf{CW}_p^p(n, m)$ of compatible PCs rooted at nodes n, m

Note: We omit calls to a cache storing previously-computed coupling circuits COUPLE(n, m) for simplicity.

if n, m are input nodes **then**
  $r \leftarrow$ input node denoting OT plan between n, m
else if n, m are sum nodes **then**
  $r \leftarrow$ new sum node with parameters $\theta_{i,j}$
  for all $c_i \in n$.children, $c_j \in m$.children **do**
    r.children$[i, j] \leftarrow$ COUPLE(c_i, c_j)
  end for
  $$\mathsf{LP} \leftarrow \begin{cases} \min & \sum_{i,j} \mathsf{CW}_p(r.\text{children}[i,j]) * \theta_{i,j} \\ \text{s.t.} & \forall i, \sum_j \theta_{i,j} = \theta_i \\ & \forall j, \sum_i \theta_{i,j} = \theta_j \\ & \theta_{i,j} \in [0, 1] \end{cases}$$
  solve LP ▷ Solve for optimal parameters $\theta_{i,j}$
else if n, m are product nodes **then**
  $r \leftarrow$ new product node
  for all $c_1 \in n$.children, $c_2 \in m$.children **where** $\mathsf{sc}(c_1) = \mathsf{sc}(c_2)$ **do**
    add COUPLE(c_1, c_2) to r.children
  end for
end if
return r

solving a linear program. We can decompose the optimization problem this way because the optimization at children are independent of the parent parameters. At a product node, we can again push the problem down to the children: $\min_\theta \sum_{c \in \mathsf{ch}(n)} g(c) = \sum_{c \in \mathsf{ch}(n)} (\min_\theta g(c))$, because the children nodes $c \in \mathsf{ch}(n)$ have disjoint scopes due to decomposability and thus do not share any parameters.

Since the time to solve the linear program at each sum node depends only on the number of children of the sum node, which is bounded, we consider this time constant when calculating the runtime of the full algorithm. Thus, we can compute CW_p and the corresponding transport plan between two circuits in time linear in the size of the coupling circuit, or equivalently, quadratic in the size of the original input circuits. Appendix B.6 presents correctness proof of the algorithm in more detail.

Compatibility has been shown to enable tractable algorithms for many pairwise queries on PCs, including information-theoretic divergences such as the Kullback-Leibler, Jensen-Shannon, and Rényi divergences [Vergari et al., 2021]. Interestingly, while such divergences are commonly considered to be easier to compute than the Wasserstein distance (as they do not require solving an optimization problem), CW_p computation is actually *easier*, as algorithms to compute the former all require an additional structural property called *determinism*. Furthermore, while our algorithm for CW_p requires the two circuits to be compatible, any pair of arbitrary non-compatible circuits may be transformed into two structured-decomposable circuits and then made compatible—albeit with a worst-case exponential increase in circuit size [de Colnet and Mengel, 2021, Zhang et al., 2024]. Lastly, current state-of-the-art PC learning algorithms naturally allow us to learn compatible circuit structures—assuming we assign the bijective mapping ourselves [Dang et al., 2020, Liu and Van den Broeck, 2021].

3.3 RELATION TO DISTANCE BETWEEN GMMS

As probabilistic circuits with Gaussian input distributions can be interpreted as deep, compact representations of Gaussian mixture models (GMMs), existing works studying optimal transport for GMMs [Chen et al., 2018, Delon and Desolneux, 2020] are highly relevant. In particular, our proposed notion of Circuit Wasserstein distance is closely related to the Mixture Wasserstein distance (MW_2) introduced by Delon and Desolneux [2020], who also derived an upper bound on the true Wasserstein distance by restricting the coupling set to a GMM structure with quadratic number of components and computed this metric by solving a linear program.

We can directly leverage this algorithm to compute a bound on p-Wasserstein distance between PCs. Specifically, we can "unroll" PCs with Gaussian inputs into their shallow representations which correspond to GMMs and them compute MW_p between those unrolled GMMs. However, the shallow representation of a PC may be exponentially larger than the size of the original circuit, making this naive approach intractable; nevertheless, we consider this approach as a baseline for our proposed approach and provide a detailed runtime comparison in Section 5. Furthermore, we observe that MW_p will be no larger than our proposed CW_p because a coupling circuit can also be unrolled into a GMM and thus must be in the coupling set for MW_p.

4 PC PARAMETER LEARNING USING WASSERSTEIN DISTANCE

Motivated by past works that train generative models by minimizing the Wasserstein distance between the model and the empirical data distribution [Rout et al., 2022b, Salimans et al., 2018, Tolstikhin et al., 2018, Arjovsky et al., 2017], we investigate the applicability of minimizing the Wasserstein distance between a PC and data as a means of learning the parameters of a given PC structure.

Formally, suppose we have a dataset $\mathcal{D} = \{\mathbf{y}^{(k)}\}_{k=1}^n$ that induces the empirical probability measure $\hat{Q}$. Then for a given PC structure, we find its parameters θ to optimize the

following:

$$\min_{\theta} \mathsf{W}_p^p(P_\theta, \hat{Q}) = \min_{\theta} \inf_{\gamma \in \Gamma(P_\theta, \hat{Q})} \mathbb{E}_{\gamma(\mathbf{x},\mathbf{y})}[\|\mathbf{x}-\mathbf{y}\|_p^p]$$
$$= \min_{\theta} \inf_{\gamma \in \Gamma(P_\theta, \hat{Q})} \frac{1}{n} \sum_{k=1}^{n} \mathbb{E}_{\gamma(\mathbf{x}|\mathbf{y}^{(k)})} \left[\left\| \mathbf{x} - \mathbf{y}^{(k)} \right\|_p^p \right]. \quad (4)$$

Note that Equation 4 comes from rewriting $\gamma(\mathbf{x}, \mathbf{y}) = \gamma(\mathbf{x}|\mathbf{y})\gamma(\mathbf{y})$, then applying linearity of expectation since $\hat{Q}$ is an empirical distribution. Unfortunately, solving the above optimization problem is computationally hard.

Theorem 2. *Suppose P_θ is a smooth and decomposable probabilistic circuit, and $\hat{Q}$ is an empirical distribution induced by a dataset $\mathcal{D} = \{\mathbf{y}^{(k)}\}_{k=1}^n$. Then computing the parameters θ that minimizes the empirical Wasserstein distance $\mathsf{W}_p^p(P_\theta, \hat{Q})$ (i.e., solving Equation 4) is NP-hard.*

We can show the above by a reduction from k-means clustering (Appendix B.5).

4.1 WASSERSTEIN-MINIMIZATION: AN ITERATIVE ALGORITHM

We again tackle this computational hardness by imposing a circuit structure on the coupling measure, allowing us compute the Wasserstein objective and optimize it efficiently.

Definition 5 (Empirical Circuit Wasserstein distance)**.** *Let P be a PC distribution and $\hat{Q}$ an empirical distribution induced by a dataset $\mathcal{D} = \{\mathbf{y}^{(k)}\}_{k=1}^n$. The p-Empirical Circuit Wasserstein distance between P and $\hat{Q}$ is*

$$\mathsf{ECW}_p^p(P, \hat{Q}) = \min_{\gamma} \frac{1}{n} \sum_{k=1}^{n} \mathbb{E}_{\gamma(\mathbf{x}|\mathbf{y}^{(k)})} \left[\left\| \mathbf{x} - \mathbf{y}^{(k)} \right\|_p^p \right],$$

where $\gamma(\mathbf{x}, \mathbf{y}^{(k)})$ satisfies the following: (i) for each $k \in \{1, \ldots, n\}$, $\gamma(., \mathbf{y}^{(k)})$ is a PC with the same structure as P (but not necessarily the same parameters) that normalizes to $1/n$, and (ii) $\sum_{k=1}^n \gamma(\mathbf{x}, \mathbf{y}^{(k)}) = P(\mathbf{x})$.

A coupling satisfying the above structure clearly satisfies the marginal constraints and is in $\Gamma(P, \hat{Q})$. Therefore, the empirical Circuit Wasserstein distance upper-bounds the empirical Wasserstein distance: $\mathsf{W}_p(P, \hat{Q}) \leq \mathsf{ECW}_p(P, \hat{Q})$. We will thus learn the parameters of PCs by minimizing this upper bound, which can be computed efficiently as follows.

We now present our *iterative algorithm* for minimum-Wasserstein parameter learning. In particular, we wish to learn the circuit parameters θ that minimizes $\mathsf{ECW}_p^p(P_\theta, \hat{Q})$ which is in turn a minimization problem over couplings γ. Thus, we alternate between (i) optimizing the coupling given the current circuit parameters and (ii) updating the circuit parameters given the current coupling.

Let us first discuss step (i) which computes $\mathsf{ECW}_p^p(P_\theta, \hat{Q})$ for a given θ and in the process finds the corresponding coupling γ. First, rather than materializing a PC to represent $\gamma(., \mathbf{y}^{(k)})$ for each k as described in Definition 5, we can equivalently model a single coupling circuit γ as having the same structure as P and a set of parameters $\{w_{r,c,k}\}_{k=1}^n$ for each parameter $\theta_{r,c}$ in P. Then optimizing the coupling circuit parameters amounts to minimizing the expected distance according to the coupling distribution, similar to computing CW, and can be done efficiently by solving a small linear program at each sum node. Here, we have the following marginal-matching constraints: $\sum_k w_{r,c,k} = \theta_{r,c}$ for each sum node r and child c and $\sum_c w_{r,c,k} = 1/n$ for each k.

Interestingly, the above linear program at each sum node is a variation of the continuous knapsack problem [Michael Goodrich, 2002] and thus has a closed-form solution. In particular, the solution results in a coupling circuit with each weight $w_{c,k}$ being either $\frac{1}{n}$ or zero (details in Appendix B.7). Intuitively, the coupling circuit parameters w describe how each data point is routed through the circuit; because the optimal coupling is deterministic—each data point is either routed wholly through an edge or not at all—we obtain a *transport map* between the learned PC and empirical distribution.

Next, we discuss step (ii) which estimates the parameters θ of PC P from a given coupling γ. Because the coupling has the same structure as P, and its weights $\{w_{r,c,k}\}$ satisfy marginal-matching constraints, we can simply extract the PC parameters: $\theta_{r,c} = \sum_{k=1}^n w_{r,c,k}$.

The above two steps are repeated iteratively until convergence; a pseudocode for the complete algorithm is provided in Appendix A). Due to the closed-form solution of the LP, the time complexity of one iteration of our algorithm is linear in both the size of the circuit and the size of the dataset, and our algorithm is also guaranteed to converge (potentially to a local minimum) as the empirical Wasserstein objective is non-increasing in every iteration (Appendix B.8). Nevertheless, finding the global optimum parameters minimizing the Wasserstein distance is still NP-hard, and our proposed efficient algorithm may get stuck at a local minimum, similar to existing maximum-likelihood parameter learning approaches.

In an effort to avoid getting stuck at a poor local minimum, we also introduce a variant of Wasserstein-Minimization called Stochastic Wasserstein-Minimization. Simply, instead of routing each data point optimally every time, we route each data point optimally with probability $1 - p$ and randomly with probability p.

We briefly remark on interesting parallels between our proposed Wasserstein-Minimization (WM) methods and Expectation-Maximization (EM) for maximum-likelihood parameter learning. EM is an iterative algorithm that al-

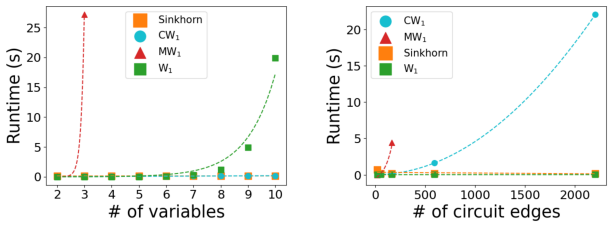

Figure 3: Runtime of Wasserstein-type distance computation using our approach (blue dots) and the baselines (MW_1 red triangles, W_1 green squares, and Sinkhorn distance with orange squares). *Left:* Fixed $k = 4$, variable v. *Right:* Fixed $v = 2$, variable k. Each data point is averaged over 20 runs. See Appendix C.1 for more detailed experimental results.

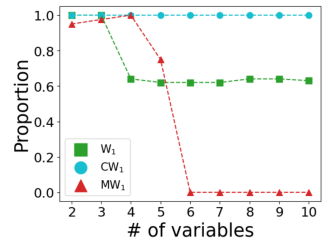

Figure 4: Proportion of instances that CW_1, MW_1, and W_1 could be solved without numerical stability or OOM issues. Note that only CW_1 could be computed exactly for every circuit pair.

ternates between (i) computing the expected likelihood (marginalizing out the latent variables) of current parameters in the E-step and (ii) estimating the parameters that maximize this in the M-step, which is analogous to the two steps of WM: (i) computing the ECW for current parameters and (ii) updating the parameters to minimize ECW.

5 EXPERIMENTS

In this section, we empirically evaluate our proposed algorithm for computing CW_p[4] against the algorithm for computing MW_p proposed by Delon and Desolneux [2020], the exact computation of W_p via solving a linear optimization problem, and an approximation of W_p via the Sinkhorn algorithm [Cuturi, 2013]—an approximate optimal transport solver between point clouds. We then compare the modeling capabilities of circuits learned using our proposed Wasserstein-Minimization (WM) algorithm against the Expectation-Maximization (EM) algorithm for PCs. Specifically, we aim to answer the following questions:

1. How does the runtime of our algorithm for CW_p scale with the size of the circuit in practice, and how does that compare to MW_p and W_p computation?
2. How effective is CW_p as a proxy metric for W_p?
3. How useful is the transport plan given by a coupling circuit?
4. How do circuits learned using our WM algorithm compare with those learned using EM?

5.1 EFFICIENCY OF CW_p COMPUTATION

To evaluate the runtime of computing CW_p against MW_p and W_p, we randomly generate pairs of circuits with Bernoulli input distributions with a given variable scope size v and sum node branching factor k. To do this, we first construct a hierarchical scope partitioning given the variable scope size, then construct two PCs following this scope partitioning (implying that they are compatible) with sum node branching factor k. These circuits mirror the structures learned using a state-of-the-art structure learning algorithm—HCLT [Liu and Van den Broeck, 2021]—where k corresponds to the block size of the HCLT structure. Estimating W_p using the Sinkhorn algorithm is done by sampling n data points from each PC and computing the Sinkhorn distance between the samples.

We implement our algorithm as detailed in Algorithm 1 to compute the optimal transport map and value for CW_p. For our baselines, (i) we implement a PC-to-GMM unrolling algorithm and employ the algorithm proposed by Delon and Desolneux [2020] to compute MW_p between the unrolled circuits, and (ii) we also compute W_p between PCs representing discrete distributions by enumerating the likelihood of every variable assignment for each circuit and solving a linear program to get the exact Wasserstein distance.

The results are summarized in Figure 3. We observe that the time to compute W_p remains effectively constant in k but grows exponentially in v, as the number of joint assignments and thus the size of the linear program grows exponentially in v. The time to compute MW_p grows exponentially in both v and k, while the time to compute CW_p grows linearly in v and quadratically in k. Therefore, CW_p is the only metric that can be efficiently computed for any circuit.

We further note that the exponential increase in problem size for MW_p and W_p introduces both numerical stability and out-of-memory issues on a machine with 256Gb of RAM when computing the metrics for larger circuits, as an exponentially larger linear optimization problem must be solved. Conversely, this is never the case for computing CW_p, as the required linear programs remain small with only k^2 variables and do not depend on the circuit size. See Figure 4 for more details.

The choice of Bernoulli leaves is intentional to allow W_p to be used as a baseline; while the choice of leaf distribution is irrelevant for MW_p and CW_p computation time, choosing a leaf distribution with even 3 categories would render most of the W_p experiments performed above impractical due to the

[4] Implementation and experiments can be found here: `https://github.com/aciotinga/pc-optimal-transport`

requirement of enumerating every variable assignment for the circuit. Furthermore, it is impossible to compute W_p in this manner using *any* continuous distribution for the leaves, while MW_p and CW_p can still be computed in the same amount of time. See Appendix C.1 for runtime experiments with circuits with Gaussian input distributions.

5.2 CW_p AS A PROXY METRIC FOR W_p

It is well-established that W_p is a useful metric for comparing distances between distributions despite it often being impractical to compute; thus, a natural question is whether this utility extends to CW_p.

CW_p Between Randomly-Generated Circuits We first evaluate CW_p between pairs of small, randomly-generated circuits for which W_p can be computed exactly in order to determine how similar our metric CW_p ranks the distances between these pairs when compared to W_p. Intuitively, the closer these rankings are, the better CW_p is as a proxy for W_p. Thus, for each of the n circuit pairs randomly generated using the same procedure as in Section 5.1, we compute two rankings—one ranking by CW_1 and the other by W_1—and compute the Kendall rank correlation coefficient τ between the two rankings. Across all settings for v and k for which W_1 was computable without running into numerical stability issues (30 out of 45 instances), the Kendall rank correlation coefficient τ was at least 0.52, with the mean of 0.70 and variance of 0.007. This indicates a strong correlation between the distance rankings of CW_1 and W_1.

We also investigate the Pearson correlation between CW_1 and W_1, and compare it to the Pearson correlation between the Sinkhorn distance and W_1. Across all settings for v and k not encountering numerical stability issues, the average correlation between CW_1 and W_1 was **0.90**, while the average correlation between the Sinkhorn distance and W_1 was **0.29** (with the highest correlation being 0.69).

CW_p Between Circuits Learned from Data To support the efficacy of our metric and algorithm on large, high-dimensional PCs beyond randomly-generated circuits, we also evaluate on PCs learned on the MNIST dataset [LeCun et al., 1998]—a 784-dimensional handwritten digits image dataset. Specifically, we first partition the dataset into 10 digit classes, then learn one PC per class using the HCLT structure learning algorithm [Liu and Van den Broeck, 2021] with a fixed block size of 4 (resulting in each circuit having over 11k edges) and EM parameter learning [Desana and Schnörr, 2016] implemented in PyJuice [Liu et al., 2024]. Between each pair of circuits, we then compute CW_1 (taking $< 5s$ per pair), the Sinkhorn distance using both 1k and 5k samples per pair (taking 1.62 and 8.51 seconds per pair, on average), as well as the average likelihood of each class-partitioned dataset for each PC. Lastly, we also compute the average *cosine similarity (CS)* between dataset partitions.

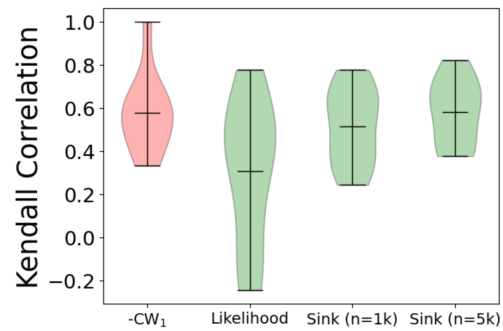

Figure 5: Distributions of Kendall correlation coefficients between Cosine Similarity and $-\mathsf{CW}_1$ (far left), likelihood (center left), negative Sinkhorn distance computed with $n = 1000$ samples (center right), and negative Sinkhorn distance computed with $n = 5000$ samples (far right). Higher is better.

Figure 6: Color transfer between images along geodesics using coupling circuits, for $t = 0, \frac{1}{3}, \frac{2}{3}, 1$ in the direction of arrows.

Figure 5 illustrates how closely correlated (according to Kendall rank correlation coefficient) the ranking by $-\mathsf{CW}_1$, negative Sinkhorn distance (computed with both 1k and 5k samples), and average likelihood are to the ranking based on average CS. We observe a significantly stronger and consistently positive correlation between CW_1 and CS rankings, when compared to that between likelihood and CS rankings. Furthermore, we observed that the variance of Sinkhorn distance is sensitive to the number of samples taken from each PC, with mean distances of 2856.2 and 2841.2 and mean standard deviations of 53.4 and 87.0 using 1k and 5k samples per pair, respectively.

5.3 CASE STUDY: COLOR TRANSFER USING CIRCUIT TRANSPORT MAPS

To evaluate the practicality of transport maps given by our coupling circuits, we adopt an application of optimal transport shown by Delon and Desolneux [2020], whereby we transport the *color histogram*—the 3-dimensional probability distribution of pixel color values—of image a to that of another image b. To do this, we learn compatible PCs $P(\mathbf{X})$ and $Q(\mathbf{Y})$ over the color distributions of images a

	EM Circuit		Deterministic WM		Stochastic WM	
k	W_2	BPD	W_2	BPD	W_2	BPD
4	32631	**1.414**	32766	1.495	**29963**	1.532
16	32873	**1.242**	32751	1.465	**29984**	1.509
64	33264	**1.192**	32749	1.458	**30999**	1.485
128	33737	**1.175**	32749	1.455	**31483**	1.474
256	34974	**1.172**	32528	1.459	**32520**	1.459

Table 1: Comparison of Wasserstein objective value and bits-per-dimension (BPD)—which is proportional to negative log-likelihood—between circuits learned via EM and two variations of WM (our approach), lower is better. The lowest value for each circuit size is bolded.

and b, compute the optimal coupling circuit $C(\mathbf{X}, \mathbf{Y})$, and transport each pixel with color value $\mathbf{x}$ to the corresponding pixel $\mathbf{y} = \mathbb{E}_C[\mathbf{Y}|\mathbf{X} = \mathbf{x}]$ (which can be computed tractably). Furthermore, we can transport each pixel value along the geodesic with points in $\mathbf{x} + t(\mathbf{y} - \mathbf{x})$ for $t \in [0, 1]$. By doing this for a fixed t for every pixel in the image, we can interpolate between two images according to our transport map in the color domain. See Figure 6 for two examples, and Appendix C.3 for additional examples.

5.4 WASSERSTEIN-MINIMIZATION FOR PARAMETER LEARNING

To determine the performance of our proposed Wasserstein-Minimization algorithm on density estimation tasks, we consider learning the parameters of circuits of various sizes from the MNIST benchmark dataset [LeCun et al., 1998]. We first generate the structure of circuits using the HCLT algorithm [Liu and Van den Broeck, 2021], varying the "block size" for different numbers of parameters. We then learn three sets of circuit parameters per structure per block size: one using mini-batch EM [Desana and Schnörr, 2016], another using our proposed deterministic WM algorithm, and a third using our stochastic WM algorithm with $p = 0.1$. All experiments were ran on a single NVIDIA L40s GPU. We refer to Appendix C.2 for more details.

Table 1 summarizes the results. We observe that our algorithms perform nearly as well as EM for learning small circuits (block size 4) benchmarked on likelihoods, and outperform EM at learning circuits with a low Wasserstein objective. However, as the size of the circuits increase, the performance of our algorithms quickly stagnates; empirically, our WM approaches do not seem to take full advantage of the larger parameter space of larger models, with models orders of magnitude larger having better but still comparable performance to their smaller counterparts when computing likelihoods and a higher Wasserstein objective.

6 CONCLUSION

This paper studied the optimal transport problem for probabilistic circuits. We introduced a Wasserstein-type distance CW_p between two PCs and proposed an efficient algorithm that computes the distance as well as associated optimal transport plan in quadratic time in the size of the input circuits, provided that they have compatible structures. We showed that CW_p always upper-bounds the true Wasserstein distance, and that—when compared to the naive application of an existing algorithm for computing a Wasserstein-type distance between GMMs to PCs—the former is exponentially faster to compute between circuits. Lastly, we propose an iterative algorithm to minimize the empirical Wasserstein distance between a circuit and data, suggesting an alternative, viable approach to parameter estimation for PCs which is often done using maximum-likelihood estimation.

We consider this work an initial stepping stone towards a deeper understanding of optimal transport theory for probabilistic circuits. Future directions include exploring more expressive formulations of coupling circuits to obtain a tighter bound on Wasserstein distance—such as relaxing the node-by-node parameter constraints to only require that the whole circuit matches marginal distributions to the original circuits. Our work also leaves open the possibility of extending the marginal-preserving properties of coupling circuits to the multimarginal setting for multimarginal generative modeling with PCs, and computing Wasserstein barycenters for PCs. Moreover, we envision that the tractable computation of a Wasserstein-type distance and transport plan between expressive models such as PCs can lead to further development in various Wasserstein-based machine learning approaches.

Acknowledgements

This work was partially supported by a gift from Cisco University Research Program. The authors would like to thank Prof. Geunyeong Byeon for helpful discussions on Wasserstein learning.

References

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`.

Emily Black, Samuel Yeom, and Matt Fredrikson. Fliptest: fairness testing via optimal transport. In *Proceedings of*

the 2020 Conference on Fairness, Accountability, and Transparency, FAT* '20, page 111–121. ACM, January 2020. doi: 10.1145/3351095.3372845. URL http://dx.doi.org/10.1145/3351095.3372845.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein Barycenters of Measures. *Journal of Mathematical Imaging and Vision*, 1(51):22–45, 2015. doi: 10.1007/s10851-014-0506-3. URL https://hal.science/hal-00881872.

Yongxin Chen, Tryphon T Georgiou, and Allen Tannenbaum. Optimal transport for gaussian mixture models. *IEEE Access*, 7:6269–6278, 2018.

Arthur Choi and Adnan Darwiche. On relaxing determinism in arithmetic circuits. In *International Conference on Machine Learning*, pages 825–833. PMLR, 2017.

YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. oct 2020. URL http://starai.cs.ucla.edu/papers/ProbCirc20.pdf.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf.

Meihua Dang, Antonio Vergari, and Guy Van den Broeck. Strudel: Learning structured-decomposable probabilistic circuits. In Manfred Jaeger and Thomas Dyhre Nielsen, editors, *Proceedings of the 10th International Conference on Probabilistic Graphical Models*, volume 138 of *Proceedings of Machine Learning Research*, pages 137–148. PMLR, 23–25 Sep 2020. URL https://proceedings.mlr.press/v138/dang20a.html.

Adnan Darwiche. A differential approach to inference in bayesian networks. *Journal of the ACM (JACM)*, 50(3): 280–305, 2003.

Sanjoy Dasgupta. The hardness of k-means clustering. UCSD Technical Report, 2008. URL https://www.cs.columbia.edu/~verma/classes/uml/ref/clustering_kmeans_NPhard_dasgupta.pdf.

Alexis de Colnet and Stefan Mengel. A compilation of succinctness results for arithmetic circuits, 2021. URL https://arxiv.org/abs/2110.13014.

Julie Delon and Agnes Desolneux. A wasserstein-type distance in the space of gaussian mixture models. *SIAM Journal on Imaging Sciences*, 13(2):936–970, 2020.

Mattia Desana and Christoph Schnörr. Expectation maximization for sum-product networks as exponential family mixture models. 04 2016. doi: 10.48550/arXiv.1604.07243.

Aden Forrow, Jan-Christian Hütter, Mor Nitzan, Philippe Rigollet, Geoffrey Schiebinger, and Jonathan Weed. Statistical optimal transport via factored couplings. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2454–2465. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/forrow19a.html.

Alison L. Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International Statistical Review / Revue Internationale de Statistique*, 70(3):419–435, 2002. ISSN 03067734, 17515823. URL http://www.jstor.org/stable/1403865.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL https://www.gurobi.com.

Leonid Kantorovich. On the transfer of masses. *Dokl. Akad. Nauk. SSSR*, 37:227–229, 1942.

Tam Le, Makoto Yamada, Kenji Fukumizu, and Marco Cuturi. Tree-sliced variants of wasserstein distances. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2d36b5821f8affc6868b59dfc9af6c9f-Paper.pdf.

Yann LeCun, Léon Bottou, Yoshua Bengio, , and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115, 2007. doi: 10.1109/ICDE.2007.367856.

Yitao Liang and Guy Van den Broeck. Towards compact interpretable models : Shrinking of learned probabilistic sentential decision diagrams. In *IJCAI 2017 Workshop on Explainable Artificial Intelligence (XAI)*, 2017. URL http://starai.cs.ucla.edu/papers/LiangXAI17.pdf.

Anji Liu and Guy Van den Broeck. Tractable regularization of probabilistic circuits. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3558–3570. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/1d0832c4969f6a4cc8e8a8fffe083efb-Paper.pdf.

Anji Liu, Kareem Ahmed, and Guy Van den Broeck. Scaling tractable probabilistic circuits: A systems perspective, 2024. URL https://arxiv.org/abs/2406.00766.

Christoph Meinel and Thorsten Theobald. *Algorithms and Data Structures in VLSI Design: OBDD-foundations and applications*. Springer Science & Business Media, 1998.

Roberto Tamassia Michael Goodrich. *Algorithm Design: Foundations, Analysis, and Internet Examples*. John Wiley & Sons, 2002.

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 689–690. IEEE, 2011.

Hamed Rahimian and Sanjay Mehrotra. Frameworks and results in distributionally robust optimization. *Open Journal of Mathematical Optimization*, 3:1–85, July 2022. ISSN 2777-5860. doi: 10.5802/ojmo.15. URL http://dx.doi.org/10.5802/ojmo.15.

Tahrima Rahman, Prasanna Kothalkar, and Vibhav Gogate. Cutset networks: A simple, tractable, and scalable approach for improving the accuracy of chow-liu trees. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14*, pages 630–645. Springer, 2014.

Litu Rout, Alexander Korotin, and Evgeny Burnaev. Generative modeling with optimal transport maps. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=5JdLZg346Lw.

Litu Rout, Alexander Korotin, and Evgeny Burnaev. Generative modeling with optimal transport maps. In *International Conference on Learning Representations*, 2022b.

Tim Salimans, Dimitris Metaxas, Han Zhang, and Alec Radford. Improving gans using optimal transport. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy den Broeck. A Compositional Atlas of Tractable Circuit Operations for Probabilistic Inference. In *Advances in Neural Information Processing Systems*, volume 34, pages 13189–13201, 2021.

Cedric Villani. *Optimal Transport, old and new*, volume 338. Springer Science & Business, 2008.

Honghua Zhang, Benjie Wang, Marcelo Arenas, and Guy Van den Broeck. Restructuring tractable probabilistic circuits, 2024. URL https://arxiv.org/abs/2411.12256.

Optimal Transport for Probabilistic Circuits (Supplementary Material)

Adrian Ciotinga[1] **YooJung Choi**[1]

[1]School of Computing and Augmented Intelligence, Arizona State University

A ALGORITHM FOR MINIMUM WASSERSTEIN PARAMETER ESTIMATION

Our proposed algorithm is broadly divided into two steps: an inference step and a minimization step. These steps are performed iteratively until model convergence. The inference step populates a cache, which stores the expected distance of each data point at each node in the circuit. This inference step is done in linear time in a bottom-up recursive fashion, making use of the cache for already-computed results. This is provided in algorithm 2.

The minimization step is done top-down recursively, and seeks to route the data at a node to its children in a way that minimizes the total expected distance between the routed data at each child and the sub-circuit. The root node is initialized with all data routed to it. At a sum node, each data point is routed to the child that has the smallest expected distance to it (making use of the cache from the inference step), and the edge weight corresponding to a child is equal to the proportion of data routed to that child; at a product node, the data point is routed to both children. Input node parameters are updated to reflect the empirical distribution of the data routed to that node. The minimization step is thus also done in linear time, and we note that this algorithm guarantees a non-decreasing objective function (see Appendix B.8 for a proof). The algorithm for this is provided in algorithm 3.

Algorithm 2 INFERENCE(n, D): returns a cache storing the distance between each data point $d_j \in D$ and each sub-circuit rooted at n, where n has children c_i. For conciseness, we omit checking for cache hits

for $c_i \in n$.children **do**
    INFERENCE(c_i, D) ▷ recursively build cache
end for
if n is a product node **then**
    for $d_j \in D$ **do**
        cache$[n, d_j] \leftarrow \sum_i$cache$[c_i, d_j]$
    end for
end if
if n is a sum node **then**
    for $d_j \in D$ **do**
        cache$[n, d_j] \leftarrow \sum_i \theta_i$cache$[c_i, d_j]$
    end for
end if
if n is an input node **then**
    for $d_j \in D$ **do**
        cache$[n, d_j] \leftarrow dist(n, d_j)$ ▷ here, $dist(n, d_j)$ is the expected distance between n and d_j
    end for
end if
return cache

Algorithm 3 LEARN(n, D, cache): learns the parameters of circuit rooted at n on data points $d_j \in D$

if not all parents of n have been learned **then**
  return ▷ We only call this method on nodes who's parents have all been learned
end if
if n is a product node **then**
  for $c_i \in n$.children **do**
    routing$[c_i] \leftarrow$ routing$[n]$ ▷ products route their data to their children
  end for
end if
if n is a sum node **then**
  $\forall \theta_i, \theta_i \leftarrow 0$ ▷ zero out parameters
  for $d_j \in$ routing$[n]$ **do**
    ▷ route data points at current node to children
    $c_i \leftarrow \arg\min_{c_i}$ cache$[c_i, d_j]$ ▷ c_i is the child node of n for which d_j has the lowest distance
    routing$[c_i] \leftarrow d_j$ ▷ route d_j to c_i
    $\theta_i \leftarrow \theta_i + \frac{1}{|\text{routing}[n]|}$ ▷ update parameter weight
  end for
end if
if n is an input node **then**
  n.parameters $\leftarrow$ parameters matching empirical distribution of routing$[n]$
end if

B PROOFS

B.1 HARDNESS PROOF OF THE ∞-WASSERSTEIN DISTANCE BETWEEN CIRCUITS

Theorem 1. *Suppose P and Q are probabilistic circuits over n Boolean variables. Then computing the ∞-Wasserstein distance between P and Q is coNP-hard, even when P and Q are deterministic and structured-decomposable.*

Proof. We will prove hardness by reducing the problem of deciding equivalence of two DNF formulas, which is known to be coNP-hard, to Wasserstein distance computation between two compatible PCs.

Consider a DNF α containing m terms $\{\alpha_1, \ldots, \alpha_m\}$ over Boolean variables $\mathbf{X}$. We will construct a PC P_α associated with this DNF as follows. For each term α_i, we construct a product of input nodes—one for each $X \in \mathbf{X}$ whose literal appears in term α_i, $\mathbb{1}[X = 1]$ for a positive literal and $\mathbb{1}[X = 0]$ for negative. Then we construct a sum unit with uniform parameters over these products as the root of our PC: $P_\alpha = \sum_{i=1}^{m} \frac{1}{m} P_{\alpha_i}$. We can easily smooth this PC by additionally multiplying P_{α_i} with a sum node $\frac{1}{2}\mathbb{1}[X = 0] + \frac{1}{2}\mathbb{1}[X = 1]$ for each variable X that does not appear in α_i. Furthermore, note that every product node in this circuit fully factorizes the variables $\mathbf{X}$, and thus the PC is trivially compatible with any decomposable circuit over $\mathbf{X}$ and in particular with any other PC for a DNF over $\mathbf{X}$, constructed as above.

Clearly, the above PC P_α assigns probability mass only to the models of α. In other words, for any $\mathbf{x} \in \{0, 1\}^n$, $P_\alpha(\mathbf{x}) > 0$ iff $\mathbf{x} \models \alpha$ (i.e. there is a term α_i that is satisfied by $\mathbf{x}$). □

B.2 PROOF OF THE MARGINAL-MATCHING PROPERTIES OF COUPLING CIRCUITS

Proposition 2. *Let P and Q be compatible PCs. Then any feasible coupling circuit C as defined in Def. 3 matches marginals to P and Q.*

Proof. We will prove this by induction. Our base case is two corresponding input nodes $n_1, n_2 \in P, Q$. The sub-circuit in C rooted at the coupling of n_1 and n_2 is an input node representing the optimal transport plan between n_1 and n_2, which clearly matches marginals to n_1 and n_2.

Now, let n_1 and n_2 be arbitrary corresponding nodes in P and Q, and assume that the coupling circuits for all children of the two nodes match marginals. We then have two cases:

Case 1: n_1, n_2 **are product nodes** Since the circuits are compatible, we know that n_1 and n_2 have the same number of children—let the number of children be k. Thus, let $c_{1,i}$ represent the i'th child of n_1, and let $c_{2,i}$ represent the i'th child of n_2. The coupling circuit of n_1 and n_2 (denoted n) is a product node with k children, where the i'th child is the coupling circuit of $c_{1,i}$ and $c_{2,i}$ (denoted c_i).

By induction, the distribution $P_{c_i}(\mathbf{X}, \mathbf{Y})$ at each child coupling sub-circuit matches marginals to the original sub-circuits: $P_{c_i}(\mathbf{X}) = P_{c_{1,i}}(\mathbf{X})$, and $P_{c_i}(\mathbf{Y}) = P_{c_{2,i}}(\mathbf{Y})$. n_1 and n_2 being product nodes means that $P_{n_1}(\mathbf{X}) = \prod_i P_{c_{1,i}}(\mathbf{X})$ and $P_{n_2}(\mathbf{Y}) = \prod_i P_{c_{2,i}}(\mathbf{Y})$, so thus $P_n(\mathbf{X}) = \prod_i P_{c_i}(\mathbf{X}) = \prod_i P_{c_{1,i}}(\mathbf{X})$ and $P_n(\mathbf{Y}) = \prod_i P_{c_i}(\mathbf{Y}) = \prod_i P_{c_{2,i}}(\mathbf{Y})$. Therefore, n matches marginals to n_1 and n_2.

Case 2: n_1, n_2 **are sum nodes** Let the number of children of n_1 be k_1 and the number of children of n_2 be k_2. Let $c_{1,i}$ represent the i'th child of n_1, and let $c_{2,j}$ represent the j'th child of n_2. The coupling circuit of n_1 and n_2 (denoted n) is a sum node with $k_1 * k_2$ children, where the (i, j)'th child is the coupling circuit of $c_{1,i}$ and $c_{2,j}$ (denoted $c_{i,j}$).

By induction, the distribution $P_{c_{i,j}}(\mathbf{X}, \mathbf{Y})$ at each child coupling sub-circuit matches marginals to the original sub-circuits: $P_{c_{i,j}}(\mathbf{X}) = P_{c_{1,i}}(\mathbf{X})$, and $P_{c_{i,j}}(\mathbf{Y}) = P_{c_{2,j}}(\mathbf{Y})$. n_1 and n_2 being sum nodes means that $P_{n_1}(\mathbf{X}) = \sum_i \theta_i P_{c_{1,i}}(\mathbf{X})$ and $P_{n_2}(\mathbf{Y}) = \sum_j \theta_j P_{c_{2,j}}(\mathbf{Y})$, so thus

$$P_n(\mathbf{X}) = \sum_i \sum_j \theta_{i,j} P_{c_{i,j}}(\mathbf{X}) = \sum_i \sum_j \theta_{i,j} P_{c_{1,i}}(\mathbf{X}) = \sum_i \theta_i P_{c_{1,i}}(\mathbf{X}) = P_{n_1}(\mathbf{X})$$
$$P_n(\mathbf{Y}) = \sum_i \sum_j \theta_{i,j} P_{c_{i,j}}(\mathbf{Y}) = \sum_i \sum_j \theta_{i,j} P_{c_{2,j}}(\mathbf{Y}) = \sum_j \theta_j P_{c_{2,j}}(\mathbf{Y}) = P_{n_2}(\mathbf{Y}) \quad (5)$$

Note that we rewrite $\sum_i \theta_{i,j} = \theta_j$ and $\sum_j \theta_{i,j} = \theta_i$ by the constraints on coupling circuits. Therefore, n satisfies marginal constraints. □

B.3 PROOF OF THE METRIC PROPERTIES OF CW_p

Proposition 1 (Metric Properties of CW_p)**.** *For any set $\mathcal{C}$ of compatible circuits, CW_p defines a metric on $\mathcal{C}$.*

Proof. It is clear that CW_p is symmetric since construction of the coupling circuit is symmetric. Furthermore, since CW_p upper-bounds W_p, it must also be non-negative.

If $\mathsf{CW}_p(P, Q) = 0$, then $\mathsf{W}_p(P, Q) = 0$ so $P = Q$. Any constraint-satisfying assignment of the parameters of a coupling circuit between P and P would also result in the Wasserstein objective at the root node being 0, since the base-case computation of W_p at the leaf nodes would always be zero.

Now, we show that CW_p satisfies the triangle inequality. Let $P, Q, R \in \mathcal{C}$ be compatible PCs over random variables $\mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$, and let $d_1 = \mathsf{CW}_p(P, Q)$, $d_2 = \mathsf{CW}_p(P, R)$, and $d_3 = \mathsf{CW}_p(R, Q)$ with optimal coupling circuits C_1, C_2, and C_3. We can construct circuits $C_2(\mathbf{x}|\mathbf{z})$ and $C_3(\mathbf{y}|\mathbf{z})$ that are still compatible with C_2 and C_3, since conditioning a circuit preserves the structure. Because all of these are compatible, we can then construct circuit $C(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = C_2(\mathbf{X}|\mathbf{Z})C_3(\mathbf{Y}|\mathbf{Z})R(\mathbf{Z})$. Thus, C is a coupling circuit of P, Q, and R such that $C_2(\mathbf{x}, \mathbf{y}) = \int C(\mathbf{x}, \mathbf{y}, \mathbf{z})d\mathbf{z}$ and $C_3(\mathbf{y}, \mathbf{z}) = \int C(\mathbf{x}, \mathbf{y}, \mathbf{z})d\mathbf{x}$. Then we have:

$$\begin{aligned}
\mathsf{CW}_p(P, Q) &= \int \|\mathbf{x} - \mathbf{y}\|_p^p C_1(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} = \int \|(\mathbf{x} - \mathbf{z}) - (\mathbf{y} - \mathbf{z})\|_p^p C(\mathbf{x}, \mathbf{y}, \mathbf{z}) d\mathbf{x} d\mathbf{y} d\mathbf{z} \\
&\leq \int \|\mathbf{x} - \mathbf{z}\|_p^p C_2(\mathbf{x}, \mathbf{z}) d\mathbf{x} d\mathbf{z} + \int \|\mathbf{z} - \mathbf{y}\|_p^p C_3(\mathbf{y}, \mathbf{z}) d\mathbf{y} d\mathbf{z} \\
&= \mathsf{CW}_p(P, R) + \mathsf{CW}_p(R, Q)
\end{aligned}$$

Thus, CW_p satisfies the triangle inequality, which concludes the proof. □

B.4 RECURSIVE COMPUTATION OF THE WASSERSTEIN OBJECTIVE

Referring to Definition 4, the Wasserstein objective for a given coupling circuit $C(\mathbf{x}, \mathbf{y})$ is the expected distance between $\mathbf{x}$ and $\mathbf{y}$. Below, we demonstrate that the Wasserstein objective at a sum node that decomposes into $C(\mathbf{x}, \mathbf{y}) = \sum_i \theta_i C_i(\mathbf{x}, \mathbf{y})$ is simply the weighted sum of the Wasserstein objectives at its children:

$$\begin{aligned}
&\mathbb{E}_{C(\mathbf{x},\mathbf{y})}[\|\mathbf{x}-\mathbf{y}\|_p^p] = \int \|\mathbf{x}-\mathbf{y}\|_p^p \, C(\mathbf{x},\mathbf{y}) d\mathbf{x} d\mathbf{y} = \int \|\mathbf{x}-\mathbf{y}\|_p^p \sum_i \theta_i C_i(\mathbf{x},\mathbf{y}) d\mathbf{x} d\mathbf{y} \\
&= \sum_i \theta_i \int \|\mathbf{x}-\mathbf{y}\|_p^p \, C_i(\mathbf{x},\mathbf{y}) d\mathbf{x} d\mathbf{y} = \sum_i \theta_i \, \mathbb{E}_{C_i(\mathbf{x},\mathbf{y})}[\|\mathbf{x}-\mathbf{y}\|_p^p] \qquad (6)
\end{aligned}$$

Now, consider a decomposable product node, where $C(\mathbf{x}, \mathbf{y}) = C_1(\mathbf{x}_1, \mathbf{y}_1) C_2(\mathbf{x}_2, \mathbf{y}_2)$ [1]. Below, we see that the Wasserstein objective at the parent is simply the *sum* of the Wasserstein objectives at its children:

$$\begin{aligned}
&\mathbb{E}_{C(\mathbf{x},\mathbf{y})}[\|\mathbf{x}-\mathbf{y}\|_p^p] = \int \|\mathbf{x}-\mathbf{y}\|_p^p \, C(\mathbf{x},\mathbf{y}) d\mathbf{x} d\mathbf{y} = \int \|\mathbf{x}-\mathbf{y}\|_p^p \, C_1(\mathbf{x}_1,\mathbf{y}_1) C_2(\mathbf{x}_2,\mathbf{y}_2) d\mathbf{x} d\mathbf{y} \\
&= \int (\|\mathbf{x}_1-\mathbf{y}_1\|_p^p + \|\mathbf{x}_2-\mathbf{y}_2\|_p^p) \times C_1(\mathbf{x}_1,\mathbf{y}_1) C_2(\mathbf{x}_2,\mathbf{y}_2) d\mathbf{x}_1 d\mathbf{x}_2 d\mathbf{y}_1 d\mathbf{y}_2 \\
&= \left(\int \|\mathbf{x}_1-\mathbf{y}_1\|_p^p \, C_1(\mathbf{x}_1,\mathbf{y}_1) d\mathbf{x}_1 d\mathbf{y}_1 \right) + \left(\int \|\mathbf{x}_2-\mathbf{y}_2\|_p^p) C_2(\mathbf{x}_2,\mathbf{y}_2) d\mathbf{x}_2 d\mathbf{y}_2 \right) \\
&= \mathbb{E}_{C_1(\mathbf{x}_1,\mathbf{y}_1)}[\|\mathbf{x}_1-\mathbf{y}_1\|_p^p] + \mathbb{E}_{C_2(\mathbf{x}_2,\mathbf{y}_2)}[\|\mathbf{x}_2-\mathbf{y}_2\|_p^p] \qquad (7)
\end{aligned}$$

Thus, we can push computation of Wasserstein objective down to the leaf nodes of a coupling circuit.

B.5 HARDNESS PROOF OF COMPUTING MINIMUM-WASSERSTEIN PARAMETERS

Theorem 2. *Suppose P_θ is a smooth and decomposable probabilistic circuit, and $\hat{Q}$ is an empirical distribution induced by a dataset $\mathcal{D} = \{\mathbf{y}^{(k)}\}_{k=1}^n$. Then computing the parameters θ that minimizes the empirical Wasserstein distance $\mathsf{W}_p^p(P_\theta, \hat{Q})$ (i.e., solving Equation 4) is NP-hard.*

Proof. We will prove this hardness result by reducing k-means clustering—which is known to be NP-hard [Dasgupta, 2008]—to learning the minimum Wasserstein parameters of a circuit. Consider a set of points $x_1 ... x_n \in \mathbb{R}^d$ and a number of clusters k. We will construct a Gaussian PC C associated with this problem as follows: the root of C is a sum node with k children; each child is a product node with d univariate Gaussian input node children (so each product node is a multivariate Gaussian comprised of independent univariate Gaussians). Minimizing the parameters of C over x_i corresponds to finding a routing of data points x_i that minimizes the total distance between all x_i's and the mean of the multivariate Gaussian child each x_i is routed to. A solution to k-means can be retrieved by taking the mean of each child of the root sum node to be the center of each of k clusters. □

B.6 PROOF OF THE OPTIMALITY OF COUPLING CIRCUIT PARAMETER LEARNING IN ALGORITHM 1

Proposition 3. *Suppose P and Q are compatible probabilistic circuits with coupling circuit C. Then the parameters of C—and thus CW_p—can be computed exactly in a bottom-up recursive fashion.*

Proof. We will construct a recursive argument showing that the optimal parameters of C can be computed exactly. Let $n \in C$ be some node in the coupling circuit C that is the coupling of nodes n_1 and n_2 in P and Q respectively. Then we have three cases:

Case 1: n is an input node By our assumption that the optimal transport plan between n_1 and n_2 can be computed in constant-time, we let n be an input node with scope $\{\mathbf{X}_k, \mathbf{Y}_k\}$ representing the optimal transport plan between n_1 and n_2, and $\mathsf{CW}_p^p(n) = \mathsf{W}_p^p(n_1, n_2)$.

[1] We assume for notational simplicity that product nodes have two children, but it is straightforward to rewrite a product node with more than two children as a chain of product nodes with two children each and see that our result still holds.

Case 2: n is a product node By recursion, $\mathsf{CW}_p^p(n) = \sum_i \mathsf{CW}_p^p(c_i)$ for each child c_i of n (see 7).

Case 3: n is a sum node Let $\theta_{i,j}$ be the parameter corresponding to the product of the i-th child of n_1 and j-th child of n_2. We want to solve the following optimization problem $\inf \mathbb{E}_{P_n(\mathbf{X},\mathbf{Y})}[\|\mathbf{X}-\mathbf{Y}\|_p^p]$, which can be rewritten as follows:

$$\inf \mathbb{E}_{P_n(\mathbf{X},\mathbf{Y})}[\|\mathbf{X}-\mathbf{Y}\|_p^p] = \inf \int_{\mathbb{R}^d\times\mathbb{R}^d} \|\mathbf{X}-\mathbf{Y}\|_p^p \, P_n(\mathbf{X},\mathbf{Y}) d\mathbf{X} d\mathbf{Y} \tag{8}$$

Rewriting the distribution of n as a mixture of its child distributions $c_{i,j}$, we get:

$$= \inf_{\theta, P_{i,j}} \int_{\mathbb{R}^d\times\mathbb{R}^d} \|\mathbf{X}-\mathbf{Y}\|_p^p \sum_{i,j} \theta_{i,j} P_{c_{i,j}}(\mathbf{X},\mathbf{Y}) d\mathbf{X} d\mathbf{Y} \tag{9}$$

Due to linearity of integrals, we can bring out the sum:

$$= \inf_{\theta, P_{i,j}} \sum_{i,j} \theta_{i,j} \int_{\mathbb{R}^d\times\mathbb{R}^d} \|\mathbf{X}-\mathbf{Y}\|_p^p \, P_{c_{i,j}}(\mathbf{X},\mathbf{Y}) d\mathbf{X} d\mathbf{Y} \tag{10}$$

Lastly, due to the acyclicity of PCs, we can separate out $\inf_{\theta_i, P_{i,j}}$ into $\inf_{\theta_i} \inf_{P_{i,j}}$ and push the latter infimum inside the sum.

$$= \inf_{\theta} \sum_{i,j} \theta_{i,j} (\inf_{P_{i,j}} \int_{\mathbb{R}^d\times\mathbb{R}^d} \|\mathbf{X}-\mathbf{Y}\|_p^p \, P_{c_{i,j}}(\mathbf{X},\mathbf{Y}) d\mathbf{X} d\mathbf{Y}) \tag{11}$$

Thus, we can solve the inner optimization problem first (corresponding to the optimization problems at the children), and then the outer problem (the optimization problem at the current node). Therefore, a bottom-up recursive algorithm is exact. □

B.7 DERIVING A CLOSED-FORM SOLUTION TO THE LINEAR PROGRAMS FOR PARAMETER UPDATES

For a sum node s with m children $s_1 ... s_m$ and a dataset with n data points $d_1 ... d_n$ each with weight w_j, we construct a linear program with $m * n$ variables $x_{i,j}$ as follows:

$$\min \quad \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbb{E}_{s_i}[\|\mathbf{X}-d_j\|_2^2] x_{i,j} \qquad \text{s.t.} \quad \sum_{i=1}^{m} x_{i,j} = w_j \; \forall j$$

Note that the constraints do not overlap for differing values of j. Thus, we can break this problem up into n smaller linear programs, each with the following form:

$$\min \quad \sum_{i=1}^{m} \mathbb{E}_{s_i}[\|\mathbf{X}-d_j\|_2^2] x_{i,j} \qquad \text{s.t.} \quad \sum_{i=1}^{m} x_{i,j} = w_j$$

The only constraint here requires that the sum of objective variables is equal to w_j. Thus, the objective is minimized when $x_{i,j}$ corresponding to the smallest coefficient takes value w_j and all other variables take value 0. Thus, the solution to the original linear program can be thought of as assigning each data point to the sub-circuit that has the smallest expected distance to it.

B.8 PROOF THAT THE WASSERSTEIN MINIMIZATION ALGORITHM HAS A MONOTONICALLY DECREASING OBJECTIVE

Proposition 4. *For a circuit rooted at n and dataset D routed to it, the Wasserstein distance between the empirical distribution of D and sub-circuit rooted at n will not increase after an iteration of algorithm A*

Proof. Let $\mathbb{E}_n[D]$ denote the Wasserstein distance between the empirical distribution of D and sub-circuit rooted at n before an iteration of algorithm A, and let $\mathbb{E}_{n'}[D]$ denote the distance after an iteration. We will show by induction that $\mathbb{E}_{n'}[D] \leq \mathbb{E}_n[D]$. Our base case is when n is an input node. By setting the parameters of n to as closely match the empirical distribution of D as possible, there is no parameter assignment with a lower Wasserstein distance to D so thus one iteration of algorithm A does not increase the objective value.

Recursively, we have two cases:

Case 1: n is a product node By the decomposition of the Wasserstein objective, we have that $\mathbb{E}_n[D] = \sum_i \mathbb{E}_{c_i}[D]$, which is $\geq \sum_i \mathbb{E}_{c_i'}[D] = \mathbb{E}_{n'}[D]$ by induction.

Case 2: n is a sum node By the decomposition of the Wasserstein objective, we have that $\mathbb{E}_n[D] = \sum_i \theta_i \, \mathbb{E}_{c_i}[D_i]$ (where $D_i \subseteq D$ is the data routed to n_i), which is $\geq \sum_i \theta_i \, \mathbb{E}_{c_i'}[D_i] = \mathbb{E}_{n'}[D]$ by induction. Our parameter updates also update each $D_i \to D_i'$, but that also guarantees that $\mathbb{E}_{c_i'}[D_i] \geq \mathbb{E}_{c_i'}[D_i']$ since $D_i = D_i'$ is within the feasible set of updates for D_i. Thus, $\mathbb{E}_n[D] \geq \mathbb{E}_{n'}[D]$, so therefore the Wasserstein objective is monotonically decreasing. □

C ADDITIONAL EXPERIMENTAL RESULTS

C.1 RUNTIME EXPERIMENTS FOR COMPUTING CW_p

The value obtained for each circuit size and variable scope size is averaged over 20 runs, and we omit data points for experiments that ran out of memory. Lastly, all experiments were ran on a machine with an Intel Core i9-10980XE CPU, 256Gb of DDR5 RAM, and a single RTX3090Ti. To solve the linear programs we used Gurobi [Gurobi Optimization, LLC, 2024], a commercial linear program solver available under academic license.

We conduct runtime experiments by randomly generating 20 pairs of circuits with a given variable scope size v and sum node branching factor—also referred to as block size for HCLT structures—k. We then compute CW_1, MW_1, and W_1 as described in Section 5. Data points not displayed either run out of memory or are infeasible due to numerical stability issues. See Figure 7 for a breakdown by circuit block size and Figure 8 for a breakdown by circuit scope size.

Lastly, we demonstrate the utility of our algorithm by computing CW_p between PCs with Gaussian input distributions—for which W_p cannot be computed simply using a linear program as done previously. See Figure 9 for the results. Note that the runtime of our algorithm is quadratic in the circuit size, while computing MW_2 has exponential runtime in the circuit size.

C.2 EMPIRICAL WASSERSTEIN PARAMETER ESTIMATION EXPERIMENTAL RESULTS

We investigated the computed Wasserstein objective and bits-per-dimension (BPD) of circuits of various sizes learned using EM and WM (our method). We found that larger circuits trained via EM have a significantly lower BPD than smaller circuits, which was not the case for circuits trained via WM. Looking at the Wasserstein objective for these circuits, we see that bpd is not directly correlated with the Wasserstein objective; circuits with a lower Wasserstein objective can have a slightly higher bpd, and vice versa.

Lastly, we consider a modification of Algorithm A that employs *stochastic routing* of data at sum nodes; succinctly, we introduce hyperparameter p that introduces a probability p that a given data point is routed randomly with uniform probability to any given child node, and a probability $1 - p$ that the data point is routed optimally as detailed in Algorithm A. When $p = 0$, we refer to this as *deterministic WM*; otherwise, we refer to the algorithm as *stochastic WM*.

In our experiments, we found that $p = 0.1$ yields the best results for minimizing the Wasserstein objective. For circuits of block size 4, we observe that this significantly decreases the Wasserstein objective without a significant change to the bits-per-dimension of the learned circuit. Over 5 random restarts, the stochastic WM algorithm resulted in a Wasserstein objective between 29947 and 29986; conversely, the deterministic WM algorithm resulted in a Wasserstein objective of 32766. However, this decrease in Wasserstein distance resulted in no decrease in bits-per-dimension for the trained models, with stochastic WM yielding circuits with BPDs of between 1.503 and 1.537. See Table 1 for more details.

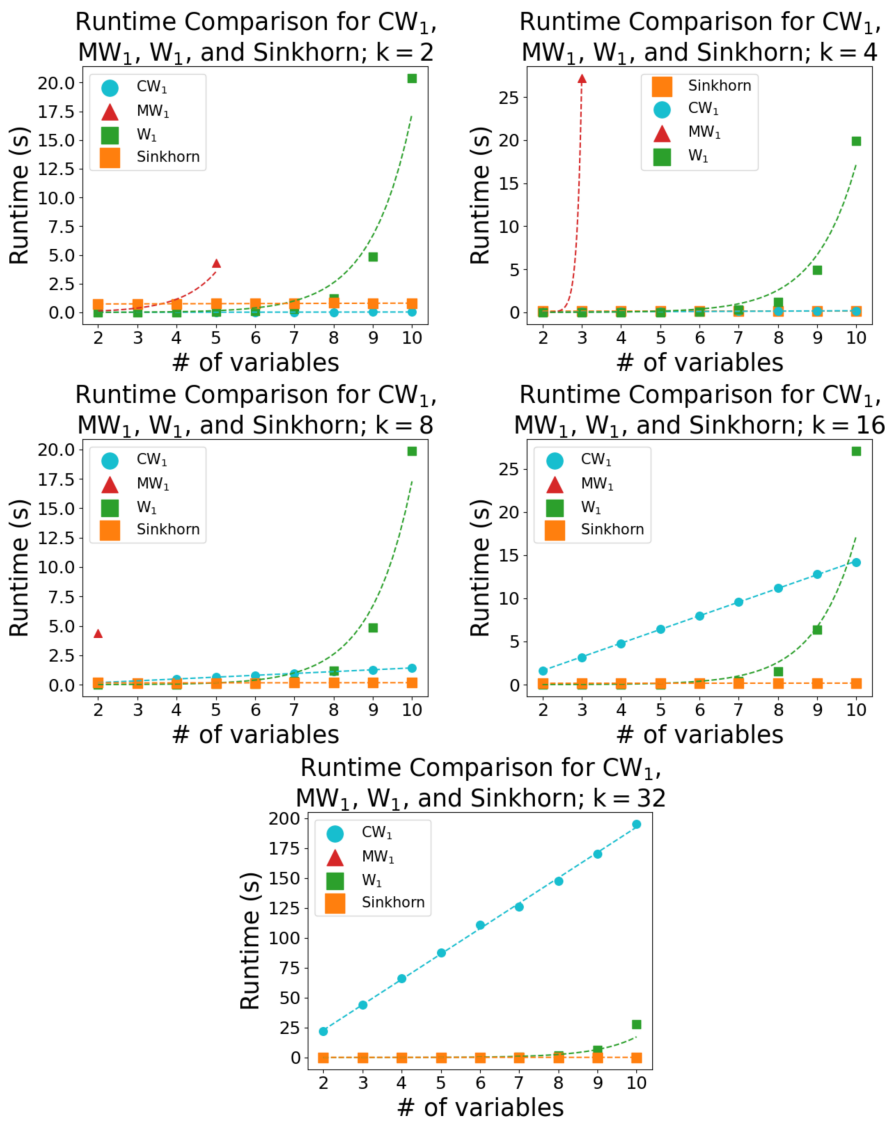

Figure 7: Runtime for algorithms computing CW_1, MW_1, and W_1. For each plot, the block size k is kept constant and the variable scope size v is varied.

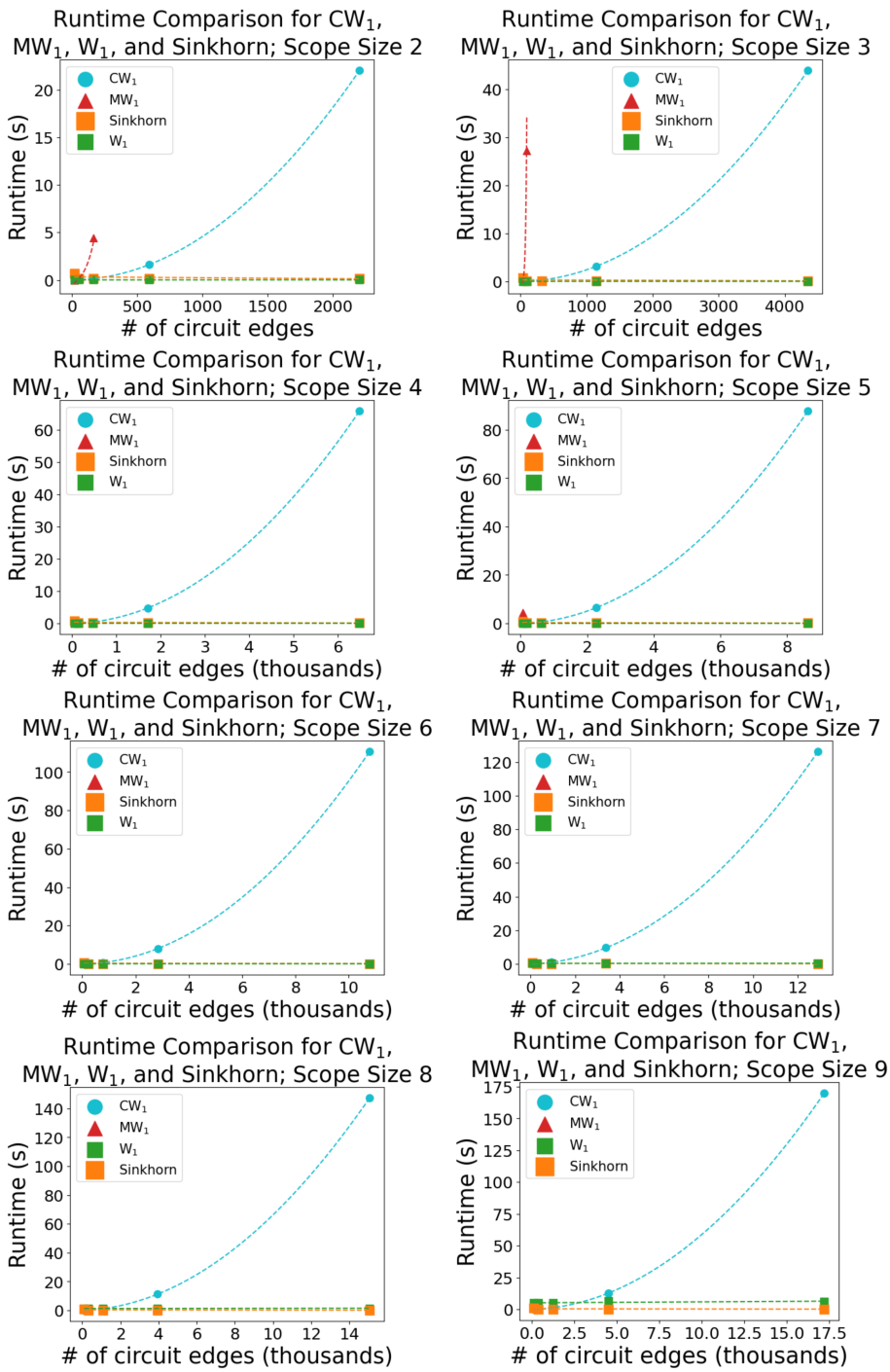

Figure 8: Runtime for algorithms computing CW_1, MW_1, and W_1. For each plot, the circuit scope size v is kept constant and the block size k is varied to adjust the number of circuit edges. The "number of circuit edges" is the number of edges in one of the two circuits.

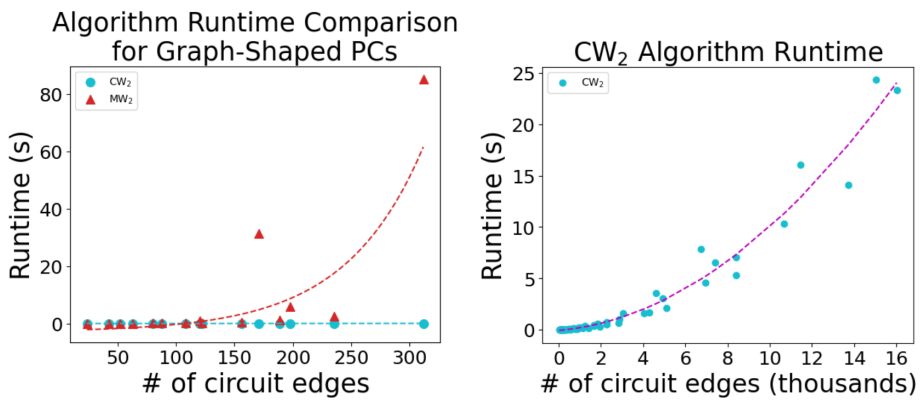

Figure 9: Runtime for algorithms computing CW_2 and MW_2 between circuits with Gaussian input distributions. As computing MW_2 quickly becomes impractical and runs out of memory, we continue with larger circuits in the second figure.

Figure 10: Color transfer between images along geodesics using coupling circuits, for $t = 0, \frac{1}{3}, \frac{2}{3}, 1$ in the direction of arrows.

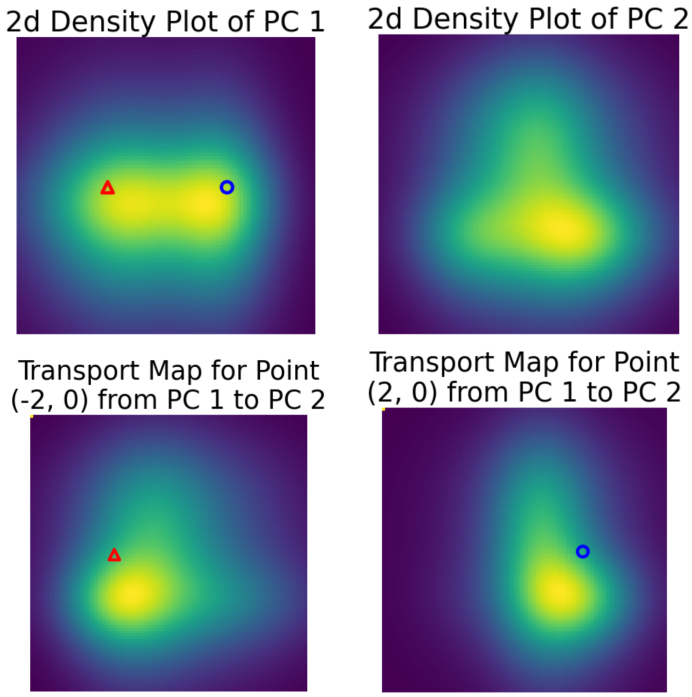

Figure 11: Visualization of transporting the indicated points from the distribution parameterized by PC 1 to the distribution parameterized by PC 2. The points (red triangle and blue circle) were arbitrarily selected to show how a point mass is redistributed according to the computed transport map. The top two figures visualize the input distributions, while the bottom two figures visualize where the point density indicated is transported to from the first to the second distribution.

C.3 ADDITIONAL COLOR TRANSFER EXPERIMENTS

An application of coupling circuits we explore in Section 5 is color transfer via optimal transport. Succinctly, we transport the *color histogram*—the 3-dimensional probability distribution of pixel color values—of image a to that of another image b by learning compatible PCs $P(\mathbf{X})$ and $Q(\mathbf{Y})$ over the color distributions of images a and b, computing the optimal coupling circuit $C(\mathbf{X}, \mathbf{Y})$, and transporting each pixel with color value $\mathbf{x}$ to the corresponding pixel $\mathbf{y} = \mathbb{E}_C[\mathbf{Y}|\mathbf{X} = \mathbf{x}]$ (which can be computed tractably). See Figure 10 for some examples.

C.4 VISUALIZING TRANSPORT PLANS BETWEEN PCS

Since our algorithm does not only return CW_p between two circuits but also the corresponding transport plan, we can visualize the transport of point densities between the two distributions by conditioning the coupling circuit on an assignment of random variables in one circuit. We can similarly visualize the transport plan for an arbitrary region in one PC to another by conditioning on the random variable assignments being within said region.

Since the transport plan for a single point (or a region of points) is itself a PC, we can query it like we would any other circuit; for example, sampling a set of corresponding points, as well as computing *maximum a posteriori*—which is tractable if the original two circuits are marginal-deterministic [Choi et al., 2020]—for the transport plan of a point corresponds to the most likely corresponding point in the second distribution for the given point. Because a coupling circuit inherits the structural properties of the original circuit, it is straightforward to understand what queries are and are not tractable for a point transport map.

In Figure 11, we provide an example of visualizing the optimal transport plan between two randomly-generated PCs. We note that despite the transport plan being constrained to be a PC with a certain structure, the resulting transport plan matches our intuition as to what an optimal transport plan should look like.