OPT2024: 16th Annual Workshop on Optimization for Machine Learning

Neural Entropic Multimarginal Optimal Transport

Dor Tsur DORTZ@POST.BGU.AC.IL
Ziv Goldfeld GOLDFELD@CORNELL.EDU
Kristjan Greenewald KRISTJAN.H.GREENEWALD@IBM.COM
Haim Permuter HAIMP@BGU.AC.IL

Abstract

Multimarginal optimal transport (MOT) is a powerful framework for modeling interactions between multiple distributions, yet its applicability is bottlenecked by a high computational complexity. Entropic regularization provides computational speedups via an extension of Sinkhorn's algorithm, whose time complexity generally scales as $O(n^k)$, for a dataset size n and k marginals. This dependence on the entire dataset size is prohibitive in high-dimensional problems that require massive datasets. In this work, we propose a new computational framework for entropic MOT (EMOT), dubbed neural entropic MOT (NEMOT), that enjoys significantly improved scalability. NEMOT employs neural networks trained using mini-batches, which transfers the computational bottleneck from the dataset size to the size of the mini-batch and facilitates EMOT computation in large problems. We provide formal theoretical guarantees on the accuracy of NEMOT via non-asymptotic error bounds that control for the associated approximation (by neural networks) and estimation (from samples) errors. We also provide numerical results that demonstrate the performance gains of NEMOT over Sinkhorn's algorithm. Consequently, NEMOT unlocks the MOT framework for large-scale machine learning.

1. Introduction

Optimal transport (OT) [32, 43] is a versatile framework for modeling complex relationships between distributions, which has achieved wide-ranging success across machine learning [3, 34, 38] statistics [11, 13], economics [18], and applied mathematics [32]. In many applications, one seeks to account for interaction between more than two marginals, which naturally leads to the multimarginal OT (MOT) setting [29]. Given $k \geq 2$ marginal distributions $\mu_1, \ldots, \mu_k$, defined on the spaces $\mathcal{X}_1, \ldots, \mathcal{X}_k$, respectively, and cost function $c : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to [0, \infty)$ the MOT problem is

$$\mathsf{MOT}_c(\mu_1, \ldots, \mu_k) \coloneqq \inf_{\pi \in \Pi(\mu_1, \ldots, \mu_k)} \int c(x_1, \ldots, x_k) d\pi(x_1, \ldots, x_k), \tag{1}$$

where $\Pi(\mu_1, \ldots, \mu_k)$ is the set of all joint distributions over $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ with ith marginal μ_i for $i = 1, \ldots, k$. MOT initially emerged as a mathematical model of certain operational problems in physics [8, 15], and more recently have found various applications in machine learning, encompassing multi-domain image translation [9], Bayesian learning [37], distributionally robust optimization [12], adversarial learning [39], fair representation learning [24] and multiview embedding matching [31]. Unfortunately, practical adoption of MOT has been limited due to its computational complexity, as it requires solving a linear program over an exponentially large (in k) number of variables. In fact, [1] shows for a variety of cost structure, a polynomial deterministic algorithm for the MOT does not exist, e.g., even if the cost tensor is low-rank.

In the bimarginal case, entropic regularization is key for the empirical successes of OT, mainly due to the computational advantages of the Sinkhorn algorithm [16, 17] and fast estimation rates [19, 26]. Entropic regularization also enables computational gains in the multimarginal setting via an extension of Sinkhorn's algorithm [23]. However, the algorithm operates on the k-wise coupling directly, which has n^k elements and results in the memory and computational costs scaling as n^k, which is exponential in the number of marginals. One approach to reduce the Sinkhorn algorithm complexity is to consider specific cost structures that benefit from a sparse graph representation (such as circle or tree structures), i.e., the k-wise cost function decomposes into a sum of a small number of pairwise costs. A cost with a sparse graph representation reduces the Sinkhorn algorithm runtime to polynomial in (n, k) [2, 4, 20]. Unfortunately, existing formulations do not lift the polynomial dependence of the overall algorithm in n, which poses a computational bottleneck when large real world datasets are considered.

1.1. Neural Entropic Multimarginal Optimal Transport

This work develops a novel methodology to scale up the EMOT computation. Drawing inspiration from recent neural estimation approaches [7, 40, 44], we propose the neural EMOT (NEMOT) framework, which optimizes an approximation of the dual form of the EMOT over a set of neural networks. This allows us to harness the vast array of deep learning tools and methodology for the task of MOT calculation while efficiently solving the optimization task using minibatch gradient descent methodology. In contrast to existing approaches, NEMOT offers increased flexibility that can handle any structure of the MOT cost. Furthermore, NEMOT can be integrated into larger systems as a regularization factor for marginals matching, e.g., multi-view learning.

To facilitate a principled implementation of NEMOT, we provide formal guarantees on its convergence given n i.i.d. samples of the of the k distributions. We show that under mild assumptions, a NEMOT implemented with m-neuron shallow ReLU networks attains a $O(\mathsf{poly}(\epsilon^{-1})k(m^{-\frac{1}{2}} + n^{-\frac{1}{2}}))$ error bound when estimating an EMOT with regularization coefficient ϵ. NEMOT provides a computationally efficient framework to estimate the MOT, as the computational bottleneck is transferred from the dataset size n to the batch size b, which is a hyper-parameter with $b \ll n$. Furthermore, we show that in popular cases where the Sinkhorn iteration complexity is polynomial in (n, k), the corresponding NEMOT computational complexity can be made polynomial in (b, k) and only linear in n/b. We then provide empirical results that demonstrate the computational advantage of NEMOT over the Sinkhorn algorithm. NEMOT's cost flexibility and data-driven approach allow for a seamless integration into modern machine learning tasks, unlocking EMOT calculation and optimization in large datasets.

1.2. Related Work

Existing applications of MOT generally aim to alter the distribution of learned representations, either by acting as a distributional constraint or matching criteria across multiple representations. In generative modeling, [9] introduce a multimarginal extension of the Wasserstein GAN [3] to address the multimarginal matching problem. In multi-view learning, the multimarginal Sinkhorn algorithm is employed for multimarginal distribution matching of learned embeddings [31]. A recent work attempted to leverage the multimarginal 1-Wasserstein distance to regularize data representations, with applications in fairness, harmonization, and generative modeling [6]. In the context of adversarial classification, [39] show the equivalence between MOT and certain adversarial multiclass

classification problems, allowing the derivation of optimal robust classification rules. In [28] a solution for high dimensional star-structured EMOT problems is proposed via diffusion processes. However, it does not generalize to other cost structures, leaving it mainly useful for barycenter problems. Despite these advances, the main limitation of existing methods is twofold: (i) most approaches are tailored to specific cost structures, limiting their general applicability [6, 28], and (ii) they often suffer from the aforementioned scalability issues, with computational complexity growing as $O(n^k)$ [31]. The goal of this work is to overcome the scalability bottleneck, so is to facilitate broader applicability of the EMOT estimation method in terms of cost structures.

2. Background

2.1. Entropic MOT

We consider a set of k marginals $\boldsymbol{\mu}^k := (\mu_1, \dots, \mu_k)$ supported on $\boldsymbol{\mathcal{X}}^k := \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ respectively. We assume in this work that $\mathcal{X}_i \subset \mathbb{R}^{d_i}$ is compact for all i. For a given cost function $c : \boldsymbol{\mathcal{X}}^k \to \mathbb{R}$ the EMOT is defined as

$$\mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k) := \inf_{\pi \in \Pi(\boldsymbol{\mu}^k)} \int c(\mathbf{x}^k) d\pi(\mathbf{x}^k) + \epsilon \mathsf{D}(\pi \| \otimes_{i=1}^k \mu_i), \tag{2}$$

where $\mathbf{x}^k = (x_1, \dots, x_k)$ and $\mathsf{D}(\mu \| \nu) := \mathbb{E}_\mu[\log(d\mu/d\nu)]$ is the KL divergence between probability measures μ and ν and $\Pi(\boldsymbol{\mu}^k)$ is the set of joint distributions over $\boldsymbol{\mathcal{X}}^k$ with marginal μ_i for every i. EMOT serves as a convexification of the linear MOT program (1) that can be solved faster but at the price of an approximation gap, which is upper bounded by $O(\epsilon \log(1/\epsilon))$ [27]. Convex duality further yield the following dual representation of EMOT

$$\mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k) = \sup_{\varphi_1, \dots, \varphi_k} \sum_{i=1}^k \int_{\mathcal{X}_i} \varphi_i d\mu_i - \epsilon \int_{\times_{i=1}^k \mathcal{X}_i} e^{(\bigoplus_{i=1}^k \varphi_i - c)/\epsilon} d(\otimes_{i=1}^k \mu_i) + \epsilon, \tag{3}$$

where $\bigoplus_{i=1}^k \varphi_i(\mathbf{x}^k) := \sum_{i=1}^k \varphi_i(x_i)$ is the direct sum of the dual potentials $\varphi_i : \mathcal{X}_i \to \mathbb{R}$, $i = 1, \dots, k$. The dual formulation is an unconstrained concave optimization w.r.t. $(\varphi_1, \dots, \varphi_k)$, and the optimal solution k-tuple is unique up to additive constants. The unique EMOT plan can be represented in terms of the dual potentials as

$$d\pi_\epsilon = \exp\left(\frac{\bigoplus_{i=1}^k \varphi_i - c}{\epsilon}\right) d(\otimes_{i=1}^k \mu_i). \tag{4}$$

The EMOT is often solved using the multimarginal Sinkhorn algorithm [23], which generalizes the matrix scaling procedure [33] to high dimensional data structures. The computational bottleneck of the Sinkhorn algorithm follows a marginalization step of the coupling tensor, whose complexity is in general $O(n^k)$. This is a key limitation that this paper aims to overcome.

2.2. Graphical Representation of the Cost

We consider a class of MOT cost functions that decompose into pairwise terms,[1] i.e., $c(\mathbf{x}^k) = \sum_{(i,j)\in\mathcal{E}} \tilde{c}(x_i, x_j)$ This enables representing the cost using a graph $\mathcal{G}_c = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} =$

1. For simplicity of presentation we consider use the same pairwise cost function $\tilde{c}$ for all pairs, note that the framework readily adapts to edge-dependent pairwise cost functions.

$\{1, \dots, k\}$ and the edge set $\mathcal{E}$ corresponds to the pairwise components on the decomposition Namely, $(i, j) \in \mathcal{E}$ if the term $\tilde{c}(x_i, x_j)$ appears in the cost decomposition. The size of $\mathcal{V}$ is induced by the structure of $c(\mathbf{x}^k)$ and determines the complexity of the coupling marginalization. In the general case, where $\mathcal{V}$ considers all pair $i \neq j$, $|\mathcal{V}|$ grows as k^2 and the corresponding Sinkhorn iteration complexity remains $O(n^k)$. We refer to such $c(\mathbf{x}^k)$ as a *full cost*. Imposing structure on the graph can reduce the scaling with k, as the coupling marginalization can be reformulated as a set of matrix-matrix operations [2, 4, 20]. For instance, in circular and tree graphs, $|\mathcal{V}|$ grows linearly with k and the Sinkhorn complexity is reduced to polynomial in (n, k).

Algorithm 1: NEMOT

Input: Dataset $\mathbf{X}^{n,k}$
Output: Optimized dual NNs
initialize k dual NN parameters $(\theta_i)_{i=1}^k$, $\ell = 0$;
while *not converged* **do**
  Sample batch $\mathbf{X}^{b,k}$;
  Calculate $\boldsymbol{f}_i = [f_{\theta_i(x_{i,j})}]_{j=1}^b$ for $i = 1, \dots, k$;
  Calculate NEMOT loss $\widehat{\mathsf{MOT}}_{c,\epsilon}(\mathbf{X}^{b,k})$;
  Update parameters:;
  $\theta_\ell \leftarrow \theta_\ell + \nabla_{\theta_\ell} \widehat{\mathsf{MOT}}_{c,\epsilon}(\mathbf{X}^{b,k})$;
  $\ell \leftarrow \ell + 1$
end
Return $\widehat{\mathsf{MOT}}_{c,\epsilon}(\boldsymbol{X}^{n,k})$, neural plan $\Pi^{\epsilon}_{\theta^\star}(\mathbf{X}^{n,k})$

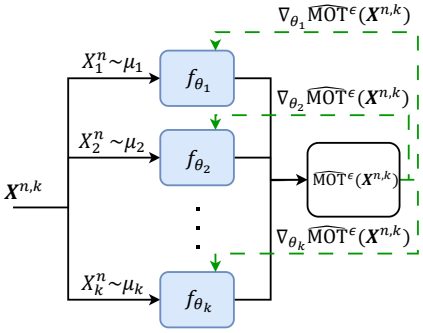


Figure 1: NEMOT system. Each marginal component of $\mathbf{X}^{n,k}$ is fed to the corresponding neural potential $(f_{\theta_i})_{i=1}^k$ and outputs are fed into the EMOT loss (5), from which gradients are calculated.

3. Neural Estimation of Entropic MOT

We propose a neural EMOT algorithm as a means to further scale up computation to massive datasets. We propose to parameterize the set of dual potentials with neural networks, approximate expectations with sample means, and optimize the resulting objective over the space of neural network (NN) parameters. Given $(n \times k)$ i.i.d. samples $\mathbf{X}^{n,k} := (X_{i,j})_{i=1,j=1}^{k,n}$ where $(X_{i,1}, \dots, X_{i,n}) \overset{i.i.d.}{\sim} \mu_i$. NEMOT then takes the form

$$\widehat{\mathsf{MOT}}_{c,\epsilon}(\mathbf{X}^{n,k}) := \sup_{\theta_1,\dots,\theta_k} \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{k} f_{\theta_i}(X_{i,j}) - \frac{\epsilon}{n^k} \sum_{j_1,\dots,j_k} e^{\left(\sum_{i=1}^k f_{\theta_i}(X_{j_i}) - c(X_{j_1},\dots,X_{j_k})\right)/\epsilon} + \epsilon. \quad (5)$$

where $f_{\theta_i} : \mathcal{X}_i \to \mathbb{R}$ is a neural network with parameters $\theta_i \in \mathbb{R}^{d'_i}$. To this end, as a random minibatch of $\boldsymbol{\mu}^k$ preserves the expectation of the objective function, the neural estimation procedure of $\mathsf{MOT}^{\epsilon}(\boldsymbol{\mu}^k)$ considers an optimization of (5) using minibatch gradient descent. The neural estimation procedure operates on the parameter space specifying the neural dual potentials.[2] The NEMOT is depicted by Figure 1. Having solved (5), we may use (4) to obtain a neural EMOT plan

$$d\pi^{\theta^\star}_{\epsilon} := e^{\left(\bigoplus_{i=1}^k f_{\theta_i^\star} - c\right)/\epsilon} d(\otimes_{i=1}^k \mu_i), \quad (6)$$

which serves to approximate the optimal solution to $\mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k)$

2. One may reduce the number of NNs by using the (c, ϵ)-transform to represent the ith network in terms of the rest.

Remark 1 (Complexity reduction for sparse cost graphs) *The naive implementation of minibatch gradient descent on (5) has computational cost $O(b^{k-1}n)$ per epoch, vs. the $O(n^k)$ cost of each Sinkhorn iteration where b is the batch size. In Appendix A we show that if the cost c has graph $\mathcal{G}_c$ that is sparse, then the second term in (5) can be further accelerated to $O(kb^2n)$ computations per epoch, i.e. linear in k and n for constant b.*

3.1. Performance Guarantees

We provide a non-asymptotic bound on the NEMOT estimation error, which stems from three sources of error: (i) function approximation of the dual MOT potential; (ii) empirical estimation of the means; and (iii) optimization, which comes from employing suboptimal (e.g., gradient-based) routines. Our analysis provides sharp non-asymptotic bounds on the errors of type (i) and (ii), leaving the account of the optimization error for future work. To that end we assume for any i, $\mathcal{X}_i \subseteq [-1,1]^d$.[3] the pairwise cost $\tilde{c}$ is continuously differentiable. Many transportation costs of interest adhere to this assumption, e.g., the p-Wasserstein cost for $p > 1$. We focus on a class of m-neuron shallow ReLU networks. Define $\mathcal{F}^m_{\mathsf{nn}}(a)$ as the class of NNs $f : \mathbb{R}^m \to \mathbb{R}$, $f(z) = \sum_{i=1}^m \beta_i \phi\left(\langle w_i, z\rangle + b_i\right) + \langle w_0, z\rangle + b_0$, whose parameters satisfy $\max_{1\le i\le m} \|w_i\|_1 \vee |b_i| \le 1$, $\max_{1\le i\le m} |\beta_i| \le \frac{a}{2m}$, and $|b_0|, \|w_0\|_1 \le a$, where $\phi(z) = z \vee 0$ is the ReLU activation and a specifies the parameter bounds.The bound is given as follows

Proposition 2 (Neural Estimation Error) *There exists a constant $C > 0$ depending only on $d := \max_{i=1,\dots,k} d_i$ such that setting $a = C(1+\epsilon^{-s})$ with $s = \lfloor d \rfloor + 3$, we have*

$$\mathbb{E}\left[\left|\widehat{\mathsf{MOT}}^{m,a}_{c,\epsilon}(\mathbf{X}^{k\times n}) - \mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k)\right|\right] \lesssim_d (1+\epsilon^{-(\frac{d}{2}+1)})k^{p(\mathcal{G}_c)}(m^{-\frac{1}{2}} + n^{-\frac{1}{2}}), \tag{7}$$

where $p(\mathcal{G}_c) = d+6$ when $\mathsf{deg}(\mathcal{G}_c) = rk$ for some $r \in \mathbb{R}$ and $p(\mathcal{G}_c) = 1$ otherwise.

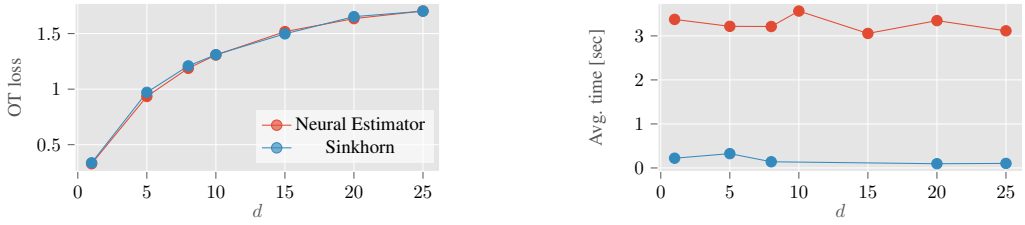
The proof of Proposition 2 is given in Appendix B.1. It decomposes the error into its approximation and estimation terms. The former is controlled by showing that the optimal dual potentials belong to some Hölder class of arbitrarily large smoothness, whence we may appeal to approximation error bounds [22]. For the estimation error, we use standard maximal inequalities from empirical process theory. The analysis is inspired by approach from [44] for bi-variate EOT neural estimation, and generalizes it to k-marginals. Proposition 2 readily extends to approximation via sigmoidal NNs, with minor adjustments to certain parameters, by leveraging the results of [5].

4. Numerical Results

In this section, we numerically demonstrate the scalability advantages of NEMOT over the baseline Sinkhorn algorithm. We consider the calculation of the EMOT between k uniform distributions over a d-dimensional normalized cube, i.e., $\mu_i = \mathsf{Unif}\left([-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}]^d\right)$. We first consider a full cost graph with pairwise quadratic cost terms. In this setting, the MOT vanishes, while the EMOT does not due to the bias created by the regularization. Additional implementation details are given in Appendix C, and the code implementation can be found in the Github repository. We compare the performance of NEMOT with that of the Sinkhorn algorithm, averaging over 5 different seeds.

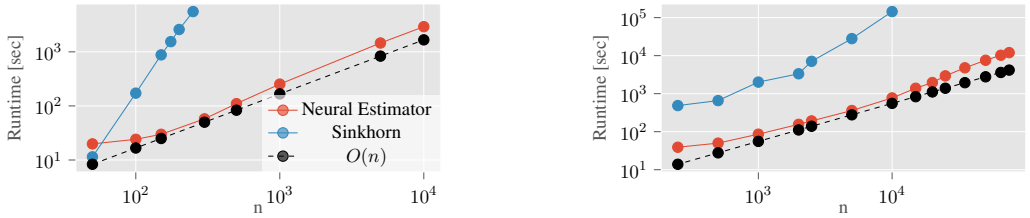
3. The considered results readily extend to arbitrary compact spaces

First, we observe in Figure 2(a) that NEMOT successfully estimates the ground truth EMOT value for a wide range of data dimension values. We further observe in Figure 2(b) that the data dimensionality does not have a significant effect on the runtime of either algorithm. Second, we study the effect of the size of the data set on the runtime of the algorithm. We observe in Figure 3(a) that, while the runtime of the Sinkhorn algorithm is exponential in the dataset size, while NEMOT remains linear in n. Then, we compare the performance of both algorithms under the computational gain of a circle cost graph. As shown in Figure 3(b), even under sparse cost graphs the NEMOT is significantly faster than Sinkhorn. Further results and analysis are given in Appendix D.



(*a*) Estimated value. (*b*) Runtime versus dimension

Figure 2: Sinkhorn vs. neural estimator on Uniform data, full cost. Note that both algorithms successfully recover the EMOT and neither runtime is bottlenecked by dimension.



(*a*) Full cost, $k = 4$ (*b*) Circle cost, $k = 10$

Figure 3: Runtime Comparison Between Sinkhorn and NEMOT under both full and circle cost structures. Note that NEMOT is significantly faster, even for the simpler circle cost structure.

5. Conclusion

This paper proposed NEMOT, a new EMOT estimation algorithm that utilizes modern NN optimization schemes. The NEMOT is cost flexible and operates in a data-driven manner, rendering it applicable for the optimization of modern machine learning models. We derive formal guarantees on the NEMOT estimation error under mild smoothness assumptions on the class of dual potentials. The resulting error are parametric (and therefore minimax optimal). We claim that NEMOT is a proper candidate to replace the Sinkhorn algorithm for massive datasets. Compared to the Sinkhorn algorithm, the NEMOT introduces a significant speedup in terms of the dataset size, while maintaining a similar level of accuracy. This claim is then demonstrated through numerical experiments. Consequently, NEMOT unlocks EMOT estimation for a new class of real world datasets. For future work, we plan to implement the NEMOT to popular EMOT applications where the Sinkhorn is the algorithmic standard, which limits the size of applicable datasets [31, 39]. Furthermore, we plan to extend the proposed methodology to additional classes of transportation problems, such as the Gromov-Wasserstein [25] and the Fused Gromov-Wasserstein [42] distances.

References

[1] Jason M Altschuler and Enric Boix-Adsera. Hardness results for multimarginal optimal transport problems. *Discrete Optimization*, 42:100669, 2021.

[2] Jason M Altschuler and Enric Boix-Adsera. Polynomial-time algorithms for multimarginal optimal transport problems with structure. *Mathematical Programming*, 199(1):1107–1178, 2023.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[4] Fatima Antarou Ba and Michael Quellmalz. Accelerating the Sinkhorn algorithm for sparse multi-marginal optimal transport via fast fourier transforms. *Algorithms*, 15(9):311, 2022.

[5] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[6] Florian Beier, Robert Beinert, and Gabriele Steidl. Multi-marginal Gromov–Wasserstein transport and barycentres. *Information and Inference: A Journal of the IMA*, 12(4):2753–2781, 2023.

[7] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540. PMLR, 2018.

[8] Jean-David Benamou, Guillaume Carlier, and Luca Nenna. Generalized incompressible flows, multi-marginal transport and Sinkhorn algorithm. *Numerische Mathematik*, 142:33–54, 2019.

[9] Jiezhang Cao, Langyuan Mo, Yifan Zhang, Kui Jia, Chunhua Shen, and Mingkui Tan. Multi-marginal wasserstein gan. *Advances in Neural Information Processing Systems*, 32, 2019.

[10] Guillaume Carlier and Maxime Laborde. A differential approach to the multi-marginal Schrödinger system. *SIAM Journal on Mathematical Analysis*, 52(1):709–717, 2020.

[11] Guillaume Carlier, Victor Chernozhukov, and Alfred Galichon. Vector quantile regression: an optimal transport approach. 2016.

[12] Louis Chen, Will Ma, Karthik Natarajan, David Simchi-Levi, and Zhenzhen Yan. Distributionally robust linear and discrete optimization with marginals. *Operations Research*, 70(3): 1822–1834, 2022.

[13] Victor Chernozhukov, Alfred Galichon, Marc Hallin, and Marc Henry. Monge–kantorovich depth, quantiles, ranks and signs. 2017.

[14] Gregory Constantine and Thomas Savits. A multivariate faa di bruno formula with applications. *Transactions of the American Mathematical Society*, 348(2):503–520, 1996.

[15] Codina Cotar, Gero Friesecke, and Claudia Klüppelberg. Density functional theory and optimal transportation with coulomb cost. *Communications on Pure and Applied Mathematics*, 66(4):548–599, 2013.

[16] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[17] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[18] Alfred Galichon. *Optimal transport methods in economics*. Princeton University Press, 2018.

[19] Aude Genevay, Lénaic Chizat, Francis Bach, Marco Cuturi, and Gabriel Peyré. Sample complexity of Sinkhorn divergences. In *The 22nd international conference on artificial intelligence and statistics*, pages 1574–1583. PMLR, 2019.

[20] Isabel Haasler, Rahul Singh, Qinsheng Zhang, Johan Karlsson, and Yongxin Chen. Multimarginal optimal transport and probabilistic graphical models. *IEEE Transactions on Information Theory*, 67(7):4647–4668, 2021.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Jason M Klusowski and Andrew R Barron. Approximation by combinations of relu and squared relu ridge functions with ℓ^1 and ℓ^0 controls. *IEEE Transactions on Information Theory*, 64(12):7649–7656, 2018.

[23] Tianyi Lin, Nhat Ho, Marco Cuturi, and Michael I Jordan. On the complexity of approximating multimarginal optimal transport. *Journal of Machine Learning Research*, 23(65):1–43, 2022.

[24] Ronak Mehta, Jeffery Kline, Vishnu Suresh Lokhande, Glenn Fung, and Vikas Singh. Efficient discrete multi-marginal optimal transport regularization. 2023.

[25] Facundo Mémoli. Gromov–Wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11:417–487, 2011.

[26] Gonzalo Mena and Jonathan Niles-Weed. Statistical bounds for entropic optimal transport: sample complexity and the central limit theorem. *Advances in neural information processing systems*, 32, 2019.

[27] Luca Nenna and Paul Pegon. Convergence rate of entropy-regularized multi-marginal optimal transport costs. *Canadian Journal of Mathematics*, pages 1–21, 2023.

[28] Maxence Noble, Valentin De Bortoli, Arnaud Doucet, and Alain Durmus. Tree-based diffusion schrödinger bridge with applications to Wasserstein barycenters. *Advances in Neural Information Processing Systems*, 36, 2024.

[29] Brendan Pass. Multi-marginal optimal transport: theory and applications. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1771–1790, 2015.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[31] Zoe Piran, Michal Klein, James Thornton, and Marco Cuturi. Contrasting multiple representations with the multi-marginal matching gap. *arXiv preprint arXiv:2405.19532*, 2024.

[32] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[33] Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967.

[34] Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (ToG)*, 34(4):1–11, 2015.

[35] Sreejith Sreekumar and Ziv Goldfeld. Neural estimation of statistical divergences. *Journal of Machine Learning Research*, 23(126):1–75, 2022.

[36] Sreejith Sreekumar, Zhengxin Zhang, and Ziv Goldfeld. Non-asymptotic performance guarantees for neural estimation of f-divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 3322–3330. PMLR, 2021.

[37] Sanvesh Srivastava, Cheng Li, and David B Dunson. Scalable bayes via barycenter in Wasserstein space. *Journal of Machine Learning Research*, 19(8):1–35, 2018.

[38] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[39] Nicolás García Trillos, Matt Jacobs, and Jakwang Kim. The multimarginal optimal transport formulation of adversarial multiclass classification. *Journal of Machine Learning Research*, 24(45):1–56, 2023.

[40] Dor Tsur, Ziv Aharoni, Ziv Goldfeld, and Haim Permuter. Neural estimation and optimization of directed information over continuous spaces. *IEEE Transactions on Information Theory*, 2023.

[41] Aad W Van Der Vaart, Jon A Wellner, Aad W van der Vaart, and Jon A Wellner. *Weak convergence*. Springer, 1996.

[42] Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Fused Gromov-Wasserstein distance for structured objects. *Algorithms*, 13(9):212, 2020.

[43] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[44] Tao Wang and Ziv Goldfeld. Neural estimation of entropic optimal transport. *arXiv preprint arXiv:2405.06734*, 2024.

Appendix A. Computational Complexity of the Neural Estimator

As NEMOT is optimized using mini-batches, it transfers the exponential time complexity from the dataset size n to the batch size b. Specifically, the complexity of a single epoch of NEMOT is $O(b^{k-1}n)$, while a single Sinkhorn iteration takes $O(n^k)$ time to compute. Furthermore, as we empirically demonstrate below, NEMOT provides good results with standard batch sizes (e.g., $b = 32, 64$), which suggests that the mini-batch may not need to scale with the dataset size. In this regimen, where $n \ll b$, we can treat b as a constant, which results in a significant speedup to the the EMOT computation.

Sparse Cost Graphs Additional computational gains are attainable when the cost graph is sparse, e.g., a circle or a tree. In such cases, the Sinkhorn complexity reduced to depend on polynomially (n, k) [2]. We next show that the batch loss expression (5) can be realized via matrix-matrix operations, which leads to a polynomial complexity in (b, k). Consequently, NEMOT maintains its computational advantages over the Sinkhorn algorithm also in the sparse cost setting. For a circle cost graph we have the following result (See Appendix B.2 for the proof).

Proposition 3 *Let $\mathcal{G}_c$ be a circle cost graph, $\boldsymbol{f}_i = [f_{\theta_i}(x_{i,1}), \ldots, f_{\theta_i}(x_{i,n})] \in \mathbb{R}^n$ be the vector of dual potential outputs, $\mathrm{C}^i[j,l] = \tilde{c}(x_{i,j}, x_{i+1,l})$ be the pairwise cost matrix for $i = 1, \ldots, k$ with $k+1 = 1$ and let $\mathrm{L}_i := \exp\left(\frac{\frac{1}{2}(\boldsymbol{f}_i \bigoplus \boldsymbol{f}_{i+1}) - \mathrm{C}_i}{\epsilon}\right) \in \mathbb{R}^{n\times n}$. Then*

$$\widehat{\mathsf{MOT}}_{c,\epsilon}(\boldsymbol{X}^{n,k}) = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{k} \boldsymbol{f}_{i,j} - \frac{\epsilon}{n^k}\mathrm{tr}\left(\prod_{i=1}^{k}\mathrm{L}_i\right) + \epsilon, \tag{8}$$

and the complexity of calculating (8) is $O(kn^3)$. Furthermore, the neural plan tensor can be represented as

$$\Pi^{\epsilon}_{\theta^\star}(\mathbf{X}^{n,k})[i_1, \ldots, i_k] = \prod_{j=1}^{k}\mathrm{L}_j(i_j, i_{j+1}). \tag{9}$$

The neural plan representation (9) allows for an efficient calculation of the unregularized OT cost induced by $\Pi^{\epsilon}_{\theta^\star}(\mathbf{X}^{n,k})$, due to the following proposition (see Appendix B.3 for the proof)

Proposition 4 *Let $(\mathrm{C}^i)_{i=1}^k$ be the pairwise cost matrices such that $\mathrm{C}^i_{j,l} = c(x_{i,j}, x_{i+1,l})$ for a given circle cost graph $\mathcal{G}_c$. The corresponding unregularized transport cost is given by*

$$\langle C, \hat{\Pi}^{\epsilon}_{\theta^\star}\rangle = \sum_{j=1}^{k}\langle C_j, m\left(\hat{\Pi}^{\epsilon}_{\theta^\star}, j, j+1\right)\rangle$$

where $m\left(\hat{\Pi}^{\epsilon}_{\theta^\star}, i_1, i_2\right)$ is the pairwise marginalization of $\hat{\Pi}^{\epsilon}_{\theta^\star}$ between (μ_{i_1}, μ_{i_2}), which is given by

$$m\left(\hat{\Pi}^{\epsilon}_{\theta^\star}, i_1, i_2\right) = \left(\prod_{j=i_1}^{i_2-1}\mathrm{L}_j\right) \odot \left[\left(\prod_{j=1}^{i_1-1}\mathrm{L}_j\right)^{\mathsf{T}}\left(\prod_{j=i_2}^{k}\mathrm{L}_j\right)^{\mathsf{T}}\right] \tag{10}$$

where $\mathrm{A} \odot \mathrm{B}$ is the Hadamard product of (A, B).

At the heart of Proposition 4 is the efficient marginalization, which is the main computational bottleneck of the Sinkhorn algorithm.

Appendix B. Proofs

B.1. Proof of Proposition 2

We begin by defining the population-level NEMOT as

$$\mathsf{MOT}^{\epsilon}_{m,a}(\boldsymbol{\mu}^k) \sup_{f_{\theta_1},\dots,f_{\theta_k}\in\times\mathcal{F}_{m_i,d_i}} \sum_{i=1}^{k}\int f_{\theta_i}d\mu_i - \epsilon\int e^{(\bigoplus_{i=1}^{k} f_{\theta_i}-c)/\epsilon}d(\otimes_{i=1}^{k}\mu_i)+\epsilon$$

First, note divide the neural estimation error into two components

$$\mathbb{E}\left[\left|\hat{\mathsf{MOT}}^{\epsilon}_{m,a}(\mathbf{X}^{k\times n}) - \mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k)\right|\right] \leq \underbrace{\mathbb{E}\left[\left|\mathsf{MOT}^{\epsilon}_{c,m,a}(\boldsymbol{\mu}^k) - \mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k)\right|\right]}_{\text{(approximation)}} + \underbrace{\mathbb{E}\left[\left|\hat{\mathsf{MOT}}^{\epsilon}_{m,a}(\mathsf{X}^{k\times n}) - \mathsf{MOT}^{\epsilon}_{c,m,a}(\boldsymbol{\mu}^k)\right|\right]}_{\text{(estimation)}},$$

where (approximation) quantifies the approximation error that is induced by replacing the optimal MOT potentials with elements from a the class of Shallow ReLU NNs and (estimation) quantifies the error the one accumulates by replacing expectations with sample means.

We begin by bounding the approximation error. Denote the optimal set of potentials with $(\varphi_1^\star,\dots,\varphi_k^\star)$.[4] The approximation error is upper bounded by given by

$$\begin{aligned}
0 &\leq \mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k) - \mathsf{MOT}^{\epsilon}_{c,m,a}(\boldsymbol{\mu}^k)\\
&\leq \sup_{f_{\theta_1},\dots,f_{\theta_k}\in\times\mathcal{F}_{m_i,d_i}} \sum_{i=1}^{k}\int_{\mathcal{X}_i}(\varphi_i^\star - f_i)\,d\mu_i\\
&\quad + \int\left(\exp\left(\frac{\sum_{i=1}^{k}\varphi_i^\star(X_i)-C(X^k)}{\epsilon}\right)-\exp\left(\frac{\sum_{i=1}^{k}f_i(X_i)-C(X^k)}{\epsilon}\right)\right)d(\otimes_{i=1}^{k}\mu_i).
\end{aligned}$$

Due to compactness of $\mathcal{X}^k$, the exponential function is consequently Lipschitz continuous on $(-\infty, M_{\mathcal{X}}]$ with Lipschitz constant $e^{M_{\mathcal{X}^k}}$, where $M_{\mathcal{X}} \coloneqq \sup_{x^k\in\mathcal{X}^k}\sum_{i=1}^{k}\varphi_i(x_i)-c(x^k)$. We therefore have

$$\begin{aligned}
\left|\mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k) - \mathsf{MOT}^{\epsilon}_{c,m,a}(\boldsymbol{\mu}^k)\right| &\leq (1+e^M/\epsilon)\sum_{i=1}^{k}\mathbb{E}_{\mu_i}\left[|\varphi_i^\star - f_i|\right]\\
&\leq (1+e^M/\epsilon)\sum_{i=1}^{k}\|\varphi_i^\star - f_i\|_{\infty,\mathcal{X}_i}\\
&\leq (\epsilon+e^M)k\epsilon^{-1}\max_{i=1,\dots,k}\|\varphi_i^\star - f_i\|_{\infty,\mathcal{X}_i} \qquad (11)
\end{aligned}$$

Due to (11), our goal boils down to bounding the sup-norm error between the optimal dual potentials and the corresponding NN approximators. To do that, we are interested in employing the following approximation bound from [36], which was previously utilized to bound the approximation error of f-divergence neural estimators by a measure of the function smoothness. To this end, we define the function class

$$\mathcal{C}_b^s(\mathcal{U}) \coloneqq \{f\in\mathcal{C}^s(\mathcal{U}) : \max_{\alpha:\|\alpha\|_1\leq s}\|D^\alpha f\|_{\infty,\mathcal{U}}\leq b\},$$

4. Existence of dual potentials is guaranteed under the assumption of finitely-supported measures [10].

where D^α, $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{Z}^d_{\geq 0}$, is the partial derivative operator of order $\sum_{i=1}^d \alpha_i$.

Proposition 5 (Proposition 10,[35]) *Let $\mathcal{X} \in \mathbb{R}^d$ be compact and let $g : \mathcal{X} \to \mathbb{R}$. Suppose that there exists an open set $\mathcal{U} \supset \mathcal{X}$, $b \geq 0$ and $\tilde{g} \in \mathcal{C}_b^{s_{\mathsf{KB}}}(\mathcal{U})$ with $s_{\mathsf{KB}} := \lfloor \frac{d}{2} \rfloor + 3$, such that $g = \tilde{g}\big|_{\mathcal{X}}$. Then, there exist $f \in \mathcal{F}_{m,d}(\bar{c}_{b,d,\|\mathcal{X}\|})$ where $\bar{c}_{b,d,\|\mathcal{X}\|}$ is a constant that depends on (b, d) and is proportional to $\max_{\|\alpha\|_1} \|D^\alpha f\|_{\infty,\mathcal{X}}$ is given in [35, Eqn. A.15], such that*

$$\|f - g\|_\infty \lesssim \bar{c}_{b,d,\|\mathcal{X}\|} d^{\frac{1}{2}} m^{-\frac{1}{2}}.$$

A preliminary step to use Proposition 5 is to that for any i, $\varphi_i \in \mathcal{C}_b^{s_{\mathsf{KB}}}$, i.e., that the function we aim to approximate adhere to the required smoothness conditions. Furthermore, for an explicit characterization of $\bar{c}_{b,d,\|\mathcal{X}\|}$ in terms of the problem's parameters, the smoothness of the partial derivatives of the dual potentials should also be characterized. To that end, we propose the following

Lemma 6 *There exist dual EMOT potentials $(\varphi_1, \dots, \varphi_k)$ for $\mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k)$ such that*

$$\max_{i=1,\dots,k} \|\varphi_i\|_{\infty,\mathcal{X}_i} \lesssim_d |\mathcal{G}_c| \tag{12}$$

$$\max_{i=1,\dots,k} \|D^\alpha \varphi_i\| \lesssim_{d,s} (1 + \epsilon^{1-s}) \left(\mathsf{deg}(\mathcal{G}_x)\right)^s, \quad 1 \leq |\alpha| \leq s. \tag{13}$$

for any $s \geq 2$ and some constant C_s that depends only on s.

The proof of Lemma 6, which is given in Appendix B.4, is a generalization of [44, Lemma 1] which accounts for the dual potential and its (c, ϵ) transform in the bimarginal EOT case. Specifically, given a set of optimal potentials $(\varphi_1^\star, \dots, \varphi_k^\star)$ we can construct a set of dual potentials $(\varphi_1', \dots, \varphi_k')$ that benefit from an explicit representation via the Schrödinger system, while agreeing with $(\varphi_1^\star, \dots, \varphi_k^\star)$ μ_k-a.s., rendering them optimal as well for the considered EMOT. Accordingly, the proof follows from similar arguments. Using Lemma 6 we result with the following error on the approximation error

$$\left|\mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k) - \mathsf{MOT}^{\epsilon}_{c,m,a}(\boldsymbol{\mu}^k)\right| \lesssim_{d,M_\mathcal{X}} \left(1 + \epsilon^{-(2+\lfloor \frac{d}{2} \rfloor)}\right) \left(\mathsf{deg}(\mathcal{G}_c)\right)^{\lfloor \frac{d}{2} \rfloor + 3} k m^{-\frac{1}{2}}, \tag{14}$$

We now turn to bound the estimator error, which is given by the following lemma

Lemma 7 (Estimation Error) *Under the considered setting, we have*

$$\mathbb{E}\left[\left|\mathsf{MOT}^{\epsilon}_{c,m,a} - \widehat{\mathsf{MOT}}^{m,a}_{c,\epsilon}(\boldsymbol{\mu}^k)\right|\right] \lesssim_d n^{-\frac{1}{2}} \max\left\{(1 + \epsilon^{-(d+4)}) \left(\mathsf{deg}(\mathcal{G}_c)\right)^{d+6}, (1 + \epsilon^{-(\frac{d}{2}+2)})k\right\}. \tag{15}$$

The proof of Lemma 7 is given in Appendix B.5 and follows the steps of the corresponding estimation error bound from [44]. It uses the established smoothness of the integrands to construct bounds on the empirical error using covering and bracketing numbers of the corresponding function classes, combined with maximum inequalities of empirical processes [41]. We note that the maximum between the two terms in (15) depends on the cost graph, as $\mathsf{deg}(\mathcal{G}_c)$ is independent on k for sparse graphs.

Finally, combining (14) and (15), we obtain

$$\begin{aligned}
&\mathbb{E}\left[\left|\widehat{\mathsf{MOT}}^{m,a}_{c,\epsilon}(\mathbf{X}^{k\times n}) - \mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k)\right|\right] \\
&\leq (1 + \epsilon^{-(\frac{d}{2}+1)}) \left(\mathsf{deg}(\mathcal{G}_c)\right)^{\frac{d}{2}+2} k m^{-\frac{1}{2}} + n^{-\frac{1}{2}} \max\left\{(1 + \epsilon^{-(d+4)}) \left(\mathsf{deg}(\mathcal{G}_c)\right)^{d+6}, (1 + \epsilon^{-(\frac{d}{2}+2)})k\right\}.
\end{aligned}$$

To obtain a simpler bound, we consider two cases, distinguishing between the the dependence of $\mathsf{deg}(\mathcal{G}_c)$ on k. In the worst case, when $\mathsf{deg}(\mathcal{G}_c) = k$, the first term is dominant in the error bound, leaving us with

$$\mathbb{E}\left[\left|\widehat{\mathsf{MOT}}^{m,a}_{c,\epsilon}(\mathbf{X}^{k\times n}) - \mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k)\right|\right] \lesssim (1+\epsilon^{-(\frac{d}{2}+1)})k^{d+6}(m^{-\frac{1}{2}}+n^{-\frac{1}{2}}),$$

while, in the sparse case where $\mathsf{deg}(\mathcal{G}_c)$ is a constant that is independent of k we have

$$\mathbb{E}\left[\left|\widehat{\mathsf{MOT}}^{m,a}_{c,\epsilon}(\mathbf{X}^{k\times n}) - \mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k)\right|\right] \lesssim (1+\epsilon^{-(\frac{d}{2}+1)})k(m^{-\frac{1}{2}}+n^{-\frac{1}{2}}).$$

This concludes the proof $\square$.

B.2. Proof of Proposition 3

Recall the the NEMOT is given by the expression

$$\widehat{\mathsf{MOT}}_{c,\epsilon}(\mathbf{X}^{n,k}) \coloneqq \sup_{\theta_1,\ldots,\theta_k} \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{k} f_{\theta_i}(X_{i,j}) - \frac{\epsilon}{n^k}\sum_{j_1,\ldots,j_k} e^{\left(\sum_{i=1}^k f_{\theta_i}(X_{j_i}) - c(X_{j_1},\ldots,X_{j_k})\right)/\epsilon} + \epsilon.$$

The first sum directly follows from the definition of the direct sum. For the exponential term, we have

$$\begin{aligned}
&\exp\left(\frac{\sum_{i=1}^k f_{\theta_i}(X_{j_i}) - c(X_{j_1},\ldots,X_{j_k})}{\epsilon}\right) \\
&\qquad = \exp\left(\frac{\sum_{i=1}^k f_{\theta_i}(X_{j_i}) - \sum_{i=1}^k \tilde{c}(X_{j_i}X_{j_{i+1}})}{\epsilon}\right) \\
&\qquad = \exp\left(\frac{\sum_{i=1}^k \frac{1}{2}\left(f_{\theta_i}(X_{j_i}) + f_{\theta_{i+1}}(X_{j_{i+1}})\right) - \sum_{i=1}^k \tilde{c}(X_{j_i}X_{j_{i+1}})}{\epsilon}\right) \\
&\qquad = \prod_{i=1}^k \exp\left(\frac{\frac{1}{2}\left(\boldsymbol{f}_{\theta_i}[j_i] + \boldsymbol{f}_{\theta_{i+1}}[j_{i+1}]\right) - \mathrm{C}^i[j_i, j_{i+1}]}{\epsilon}\right) \\
&\qquad = \prod_{i=1}^k \mathrm{L}_i[j_i, j_{i+1}].
\end{aligned}$$

Note that by the definition of the neural plan (6) we have obtained the desired representation. Consequently, the second sum is given by

$$\sum_{j_1,\ldots,j_k}\prod_{i=1}^k \mathrm{L}_i[j_i, j_{i+1}] = \sum_{j_1,\ldots,j_k} \mathrm{L}_1[j_1,j_2]\mathrm{L}_2[j_2,j_3]\cdots\mathrm{L}_k[j_k,j_1] \tag{16}$$

note that each index we sm over corresponds only to a specific pair of matrices, and is equivalent to their matrix multiplication. Thus, after taking the product of the set of matrices, we remain with the sum

$$\sum_{j_1,\ldots,j_k}\left(\prod_{i=1}^k \mathrm{L}_i\right)[j_1,j_1] = \mathrm{tr}\left(\prod_{i=1}^k \mathrm{L}_i\right)$$

completes the proof. $\square$

B.3. Proof of Proposition 4

Recall that the optimal neural plan can be represented as the multiplication of exponential matrices, i.e.,

$$\hat{\Pi}^{\epsilon}_{\theta^\star}(i_1, i_2, \dots, i_k) = \prod_{j=1}^{k} \mathrm{L}_j[i_j, i_{j+1}].$$

Let $C \in \mathbb{R}^{n^k}$ be the explicit cost tensor and $(\mathrm{C}^i)_{i=1}^k \subset \mathbb{R}^{n\times n}$ be the corresponding pairwise cost matrices. We begin with the representation of the unregularized cost. We have the following

$$\begin{aligned}
\langle \mathrm{C}, \hat{\Pi}^{\epsilon}_{\theta^\star} \rangle &= \sum_{i_1,i_2,\dots,i_k} C[i_1, i_2, \dots, i_k] \hat{\Pi}^{\epsilon}_{\theta^\star}[i_1, i_2, \dots, i_k] \\
&= \sum_{i_1,i_2,\dots,i_k} \sum_{j=1}^{k} \mathrm{C}_j[i_j, i_{j+1}] \hat{\Pi}^{\epsilon}_{\theta^\star}[i_1, i_2, \dots, i_k] \\
&= \sum_{i_1,i_2,\dots,i_k} \sum_{j=1}^{k} \mathrm{C}_j[i_j, i_{j+1}] \left(\prod_{\ell=1}^{k} \mathrm{L}_\ell[i_\ell, i_{\ell+1}] \right) \\
&= \sum_{j=1}^{k} \sum_{i_j, i_{j+1}} \mathrm{C}_j[i_j, i_{j+1}] \left(\sum_{(i_1,\dots,i_k)\setminus(i_j,i_{j+1})} \prod_{\ell=1}^{k} \mathrm{L}_\ell[i_\ell, i_{\ell+1}] \right) \\
&= \sum_{j=1}^{k} \sum_{i_j, i_{j+1}} \mathrm{C}_j[i_j, i_{j+1}] m\left(\hat{\Pi}^{\epsilon}_{\theta^\star}, j, j+1\right)[i_j, i_{j+1}] \\
&= \sum_{j=1}^{k} \langle \mathrm{C}_j, m\left(\hat{\Pi}^{\epsilon}_{\theta^\star}, j, j+1\right) \rangle.
\end{aligned}$$

We now move on to prove the circle cost marginalization formula. Let $0 \le u, v \le k$ with $u \neq v$. The marginalization follows from noticing that the considered summations over the given product can be represented with products of the matrices $\mathrm{L}_1, \dots, \mathrm{L}_k$. For a given entry of the marginalized coupling, we have the following

$$m\left(\hat{\Pi}^{\epsilon}_{\theta^\star}, u, v\right)(i_u, i_v) = \sum_{(i_1,\dots,i_k)\setminus[i_u,i_v]} \left(\prod_{\ell=1}^{k} \mathrm{L}_\ell[i_\ell, i_{\ell+1}] \right)$$

We dividethe sum into its consecutive parts and define the matrices

$$\mathrm{A}_{<u} \coloneqq \prod_{j=1}^{u-1} \mathrm{L}_j, \qquad \mathrm{A}_{u<v} \coloneqq \prod_{j=u+1}^{v-1} \mathrm{L}_j, \qquad \mathrm{A}_{>v} \coloneqq \prod_{j=v+1}^{k} \mathrm{L}_j$$

, we result with the following product:

$$\begin{aligned}
m\left(\hat{\Pi}^{\epsilon}_{\theta^\star}, u, v\right)[i_u, i_v] &= \sum_{i_1} \mathrm{A}_{<u}[i_1, i_u] \mathrm{A}_{u<v}[i_u, i_v] \mathrm{A}_{>v}[i_v, i_1] \\
&= \mathrm{A}_{u<v}(i_u, i_v) \sum_{i_1} \mathrm{A}^{\intercal}_{<u}[i_u, i_1] \mathrm{A}^{\intercal}_{>v}[i_1, i_v] \\
&= \mathrm{A}_{u<v}[i_u, i_v] \left(\mathrm{A}^{\intercal}_{<u} \mathrm{A}^{\intercal}_{>v}\right)[i_u, i_v],
\end{aligned}$$

which completes the proof. □

B.4. Proof of Lemma 6

Fix $i \in [1, \dots, k]$. We start with the bound of $\|\varphi_i^\star\|_{\infty,\mathcal{X}_i}$. By the invariance of the EMOT to additive constants we may take without loss of generality optimal MOT potentials such that $\int_{\mathcal{X}_i} \varphi_i d\mu_i = \frac{1}{k}\mathsf{MOT}_{c,\epsilon}$. We consider a set of dual potentials $(\varphi_i)_{i=1}^k$ that satisfy the Schrödinger system with the optimal potentials, i.e.,

$$\int \exp\left(\sum_{i=1}^k \varphi_i^\star(x_i) - c(x^k)\right) d\mu^{-i}(x^{-i}) = 1, \quad \mu_i - \text{a.e.},$$

where $\mathcal{X}^{-i}(\times_{j=1}^k \mathcal{X}_j) \setminus \mathcal{X}_i$ and similarly $\mu^{-i} \coloneqq \otimes_{j\in[1,\dots,k]\setminus i}\mu_j$. Thus, for any $i \in [1, \dots, k]$ we have

$$\varphi_i(x) = -\epsilon \log \int_{\mathcal{X}^{-i}} \exp\left(\frac{\sum_{-j} \varphi_j(x_j) - c(x^k)}{\epsilon} d\mu^{-i}(x^{-i})\right) \tag{17}$$

To obtain the desired bound, we derive a uniform bound over $\mathcal{X}_i$. First, by Jensen's inequality, we have

$$\begin{aligned}
\varphi_i(x_i) &\le \int_{\mathcal{X}^{-}i} c(x^k) - \sum_{j=1, j\neq i}^k \varphi_j^\star(x_j) d\mu^{-i}(x^{-i}) \\
&\le |\mathcal{G}_c| C(d) - \frac{k-1}{k}\mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k) \\
&\le |\mathcal{G}_c| C(d).
\end{aligned}$$

where the second inequality follows from the choice $\int_{\mathcal{X}_i} \varphi_i^\star d\mu_i = \frac{1}{k}\mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k) \ge 0$ and the decomposition of $c(x^k)$ into its pairwise components, each of which upper bounded $C(d_i, d_j)$, which we uniformly upper bound with $C(d)$ such that $d = \max_{i=1}^k d_i$. For example, when the pairwise cost is quadratic, $C(d_i, d_j) = 2\max(d_i, d_j)$. Next, we have

$$\begin{aligned}
\varphi_i(x_i) &\le \epsilon \log \int_{\mathcal{X}^{-}i} \exp\left(\frac{c(x^k) - \sum_{j=1, j\neq i}^k \varphi_j^\star(x_j)}{\epsilon}\right) d\mu^{-i}(x^{-i}) \\
&\le \epsilon \log \int_{\mathcal{X}^{-}i} \exp\left(\frac{(1 + \frac{k-1}{k})|\mathcal{G}_c| C(d)}{\epsilon}\right) d\mu^{-i}(x^{-i}) \\
&\le 2C(d)|\mathcal{G}_c|.
\end{aligned}$$

where, here, the second inequality follows from $\mathsf{MOT}_{c,\epsilon}(\boldsymbol{\mu}^k) \le |\mathcal{G}_c| C(d)$. As the bound is uniform we can conclude that

$$\max_{i=1,\dots,k} \|\varphi_i\|_{\infty,\mathcal{X}_i} \lesssim_d |\mathcal{G}_c|$$

Next, note that $(\phi_1, \dots, \phi_k)$ are also optimal potentials of the EMOT, as by Jensen's inequality we have

$$\begin{aligned}\sum_{i=1}^{k}(\varphi_i^\star - \varphi_i)d\mu_i &\leq \sum_{i=1}^{k}\epsilon \log \int_{\mathcal{X}_i} \exp\left(\frac{\varphi_i^\star - \varphi_i}{\epsilon}\right) d\mu_i \\ &= \sum_{i=1}^{k}\epsilon \log \int_{\mathcal{X}^k} \exp\left(\frac{\sum_{j=1}^k \varphi_j^\star - c}{\epsilon}\right) d(\otimes_{j=1}^k \mu_j) \\ &= 0.\end{aligned}$$

By the concavity of the logarithm function we can conclude that $\varphi_i = \varphi_i^\star$ μ_i-a.s. for $i = 1, \dots, k$.

Next we bound the magnitude of the partial derivative. The differentiability of the dual potentials is granted from their definition. To bound the partial derivative we will use the Faa di Bruno Formula [14, Corocllary 2.10], which for a multi-index α, provides us with the following characterization of the partial derivative of φ w.r.t. α as follows

$$\begin{aligned}-D^\alpha(\varphi_i)(x) = \epsilon \sum_{r=1}^{\alpha} \sum_{p(\alpha,r)} \frac{\alpha!(r-1)!(-1)^{r-1}}{\prod_{j=1}^{|} \alpha|(k_j!)(\beta_j!)^{k_j}} \\ \times \prod_{j=1}^{|\alpha|} \left(\frac{D^{\beta_j} \int \exp\left(\frac{\sum_{\ell=1,\ell\neq i}^k \varphi_\ell^\star(x_k) - c(x^k)}{\epsilon} d\mu^{-i}(x^{-i})\right)}{\int \exp\left(\frac{\sum_{\ell=1,\ell\neq i}^k \varphi_\ell^\star(x_k) - c(x^k)}{\epsilon} d\mu^{-i}(x^{-i})\right)} \right), \quad (18)\end{aligned}$$

where $p(\alpha, r)$ is the collection of all tuples $(k_1, \dots, k_{|\alpha|;\beta_1,\dots,\beta_{|\alpha|}}) \in \mathbb{N}^{|\alpha|} \times \mathbb{N}^{d_i \times \alpha}$ satisfying $\sum_{i=1}^{|\alpha|} k_i = r, \sum_{i=1}^{|\alpha|} k_i\beta_i = \alpha$ and for which there exit $s \in (1, \dots, |\alpha|)$ such that $k_i = 0$ and $\beta_i = 0$ for all $i \in (1, \dots, |\alpha| - s)$, $k_i > 0$ for all $i \in (|\alpha| - s + 1, \dots, |\alpha|)$ and $0 \prec \beta_{|\alpha|-s+1} \prec \cdots \prec \beta_{|\alpha|}$. We refer the reader to [14] for more information on Faa di Bruno formula. For our purpose, it is sufficient to bound $D^{\beta_j} \int \exp\left(\sum_{\ell=1,\ell\neq i}^k \varphi_\ell^\star(x_k) - c(x^k)\right) d\mu^{-i}(x^{-i})$. To that end, we apply the Faa di Bronu formula to $D^{\beta_j} \exp(-c(x^k)/\epsilon)$ to result with

$$D^{\beta_j} \exp(-c(x^k)/\epsilon) \sum_{r'=1}^{|\eta_j|} \left(\frac{-1}{2\epsilon}\right)^{r'} \sum_{p(\beta_j, r')} l(\beta_j, r', \mathbf{k}, \eta) \exp(-c(x^k) \prod_{\ell=1}^{\beta_j} D^{\eta_\ell} c(x^k),$$

and the cost derivative can be bounded using the pairwise cost bound, i.e., for any η_ℓ

$$D^{\eta_\ell} c(x^k) = \sum_{j \in N(X_i)} D^{\eta_\ell} c(x_i, x_j) \leq C'(d)\mathsf{deg}(\mathcal{G}_c).$$

where $N(X_i)$ re the neighbors of the variable X_i in the cost graph $\mathcal{G}_c$, $C'(d)$ is a uniform bound on the pairwise costs derivatives that depends only on d and $\mathsf{deg}(\mathcal{G}_c)$ is the degree of the graph $\mathcal{G}_c$. The sum $\sum_{r'=1}^{|\eta_j|} \left(\frac{-1}{2\epsilon}\right)^{r'}$ is upper bounded by $\epsilon^{-|\beta_j|}$ when $0 \leq \epsilon \leq 1$, and by ϵ^{-1} when $\epsilon > 1$. Consequently, we have uniformly on $\mathcal{X}_i$

$$|D^\alpha \varphi_i(x_i)| \leq C'(|\alpha|, d)(1 + \epsilon^{1-|\alpha|}) \left(\mathsf{deg}(\mathcal{G}_c)\right)^{|\alpha|},$$

which concludes the proof. □

B.5. Proof of Lemma 7

We start by decomposing the expression at hand. To that end we define $\mathcal{F}_{k,m}(a) = \bigcup_{i=1}^{k} \mathcal{F}_{m,d_i}(a)$. We have the following

$$\begin{aligned}
&\mathbb{E}\left[\left|\mathsf{MOT}^{\epsilon}_{c,m,a} - \hat{\mathsf{MOT}}^{\epsilon}_{a,m}(\boldsymbol{\mu}^k)\right|\right] \\
&\leq n^{-\frac{1}{2}} \sum_{i=1}^{k} \left(\mathbb{E}\left[\sup_{f_i \in \mathcal{F}_{m,d_i}(a)} \left| n^{-\frac{1}{2}} \sum_{j=1}^{n} f_i(x_{i,j}) - \mathbb{E}_{\mu_i}\left[f_i\right] \right| \right] \right) \\
&+ n^{-\frac{1}{2}} \left(\mathbb{E}\left[\sup_{(f_1,\dots,f_k) \in \mathcal{F}_{k,m}(a)} \left| n^{-\frac{1}{2}} \sum_{j=1}^{n} e^{\left(\sum_{i=1}^{k} f_i(x_{i,j}) - c(\mathbf{x}_j^k)\right)/\epsilon} - \mathbb{E}\left[e^{\left(\bigoplus_{i=1}^{k} f_i - c\right)/\epsilon} \right] \right| \right] \right).
\end{aligned}$$

Te first term contains k distinct error terms for each of the NNs f_i, $i \leq k$. This bound is therefore effectively reduced to the one considered in [44]. Consequently, we can repeat the arguments in [44, Eqn. 24], which bound the expected error using maximal inequalities of empirical processes, combined with a bound on the covering number of the dual potential function class. Thus, for each $i \leq k$ we have

$$\mathbb{E}\left[\sup_{f_i \in \mathcal{F}_{m,d_i}(a)} \left| n^{-\frac{1}{2}} \sum_{j=1}^{n} f_i(x_{i,j}) - \mathbb{E}_{\mu_i}\left[f_i\right] \right| \right] \leq a d_i^{3/2}.$$

To bound the second term we need to account for the smoothness of the exponential term we aim to estimate, which was, fortunately, already established in Appendix B.4. We follow the step in [44], with an adjustment to the exponential term smoothness bound. To avoid heavy notation, we denote $\mathsf{F}(\mathbf{x}^k) \coloneqq e^{\left(\sum_{i=1}^{k} f_i(x_{i,j}) - c(\mathbf{x}_j^k)\right)/\epsilon}$. We have

$$\begin{aligned}
\mathbb{E}\left[\sup_{(\boldsymbol{f}_{\theta,k}) \in \mathcal{F}_{k,m}(a)} \left| n^{-\frac{1}{2}} \sum_{j=1}^{n} \mathsf{F}(\mathbf{x}_j^k) - \mathbb{E}\left[\mathsf{F}\right] \right| \right] &\overset{(a)}{\lesssim} \mathbb{E}\left[\int_0^{\infty} \sqrt{\log N(\delta, \mathcal{F}_{k,m}(a), \|\cdot\|_{2,\otimes_{i=1}^{k}(\mu_i)_n})} d\delta \right] \\
&\leq \int_0^{\infty} \sqrt{\sup_{\gamma \in \mathcal{P}(\boldsymbol{\mathcal{X}}^k)} \log N(\delta, \mathcal{F}_{k,m}(a), \|\cdot\|_{2,\gamma})} d\delta \\
&\lesssim \sup_{\gamma \in \mathcal{P}(\boldsymbol{\mathcal{X}}^k)} \int_0^{12a} \sqrt{\log N(\delta, \mathcal{F}_{k,m}(a), \|\cdot\|_{2,\gamma})} d\delta \\
&\overset{(b)}{\lesssim} \sup_{\gamma \in \mathcal{P}(\boldsymbol{\mathcal{X}}^k)} \int_0^{12a} \sqrt{\log N_{[]}(\delta, \mathcal{F}_{k,m}(a), \|\cdot\|_{2,\gamma})} d\delta \\
&\overset{(c)}{\lesssim} K_s \int_0^{12a} \left(\frac{C'(s,d)(1+\epsilon^{1-s})\left(\mathsf{deg}(\mathcal{G}_c)\right)^s}{2\delta} \right)^{\frac{d}{2s}} d\delta \\
&\lesssim \tilde{C}(s,d) a (1+\epsilon^{1-s})\left(\mathsf{deg}(\mathcal{G}_c)\right)^s
\end{aligned}$$

where $\boldsymbol{f}_{\theta,k} \coloneqq (f_{\theta_1}, \dots, f_{\theta_k})$ (a) is an upper bound in terms of the function class covering number, which follows from [41, Corollary 2.8.2], since $n^{-\frac{1}{2}} \sum_{i=1}^{n} \sigma_j \mathsf{F}(\mathbf{x}_j^k)$ is Sub-Gaussian w.r.t. the pseudo metric $\|\cdot\|_{2,\otimes_{i=1}^{k}(\mu_i)_n}$ whenever $(\sigma_i)_{i=1}^{n}$ are Rademacher random variables, (b) upper bounds the covering number with the bracketing number and (c) utilizes [41, Corollary 2.7.2] which upper

bounds the bracketing number in terms of the smoothness parameter of the function class with a constant K_s that depends only in s and $C(s,d)$ is the constant from Appendix B.4.

Finally, by setting $a = \bar{c}_{b,d}$ and combining both bounds, we have

$$\mathbb{E}\left[\left|\mathsf{MOT}^{\epsilon}_{c,m,a} - \hat{\mathsf{MOT}}^{\epsilon}_{a,m}(\boldsymbol{\mu}^k)\right|\right] \lesssim_d n^{-\frac{1}{2}} \max\left\{(1+\epsilon^{-(d+4)})\left(\mathsf{deg}(\mathcal{G}_c)\right)^{d+6}, (1+\epsilon^{-(\frac{d}{2}+2)})k\right\}.$$

This concludes the proof. □

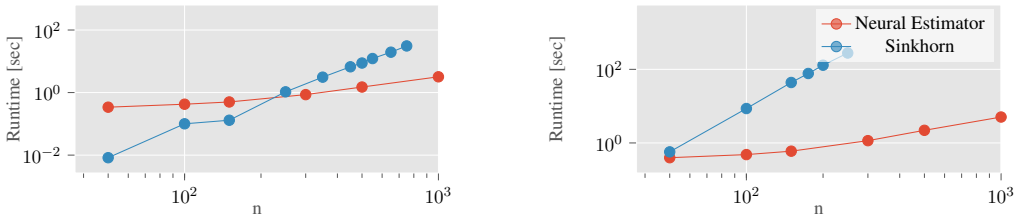
Appendix C. Additional Implementation Details

All NEMOT models are implemented in PyTorch [30]. The models are trained with the Adam optimizer [21] with an initial learning rate of 5×10^{-5}. As the neural estimation batch loss (5) considers an exponential term, we introduce a gradient norm clipping with the norm clipping parameter of 0.1. We consider an exponential learning rate decay with decay value of 0.5 and scheduling rate of 5 epochs. We consider a batch size of $b = 32$ and train the network for 50 epochs to ensure convergence of the loss for small values of n. However, in all training sessions the estimate saturated after less than 20 epochs. The NEMOT networks are implementation with simple feedfoward MLPs. The networks consists of 3 layers with output sizes $[10K, 10K, 1]$ with $K \coloneqq \min(10d, 80)$. Each layer output is followed with a ReLU activation. When calculating the neural plan, we apply normalization by the sum of its entries to very it is a proper joint distribution over the product space. In practice, we found that drawing gradients for all the k networks improved the NEMOT convergence, while at a neglible cost of runtime. This method of course introduces a trade-off, which is made crucial in the large k regime.

Appendix D. Additional Experimental Results

We present additional plots to analyse the NEMOT performance. We being by providing another outlook o the runtime comparison between the Sinkhorn and the EMOT algorithm. While the total algorithm runtime is usually the standard approach to compare runtimes, these are highly dependent on user design hyperparameters, such as error tolerance and total number of iterations. To this end, we complement the analysis in Section 4 with a comparison of the average iteration time. We compare a single Sinkhorn iteration runtime with a single NEMOT epoch, as both serve as a proxy for a pass on the entire dataset. We consider the same experimental setting of Section 4 for several combination of cost graph and k values. The results, as presented in Figure 4, show that for the full cost, when n is small ($n \leq 500$) a single Sinkhorn iteration is indeed faster than a NE epoch. However, such cases are not the ones this work aims to address, as the NEMOT is a tool designed for large n. When n is bigger, or $k > 3$, the significant computational advantage of NEMOT on the iteration level is visible.

Next, we are interested in the NEMOT runtime dependence on the data dimension and number of marginal. As seen in Figure 5, the NEMOT average iteration complexity grows approximately linearly with the number of marginals. It is also notable that the dimension have a moderate effect on the average iteration time, which stems from optimizing k NNs on each epoch, as described in Appendix C.



(a) $k = 3$ (b) $k = 4$

Figure 4: Average iteration time - Sinkhorn algorithm vs. neural estimator.

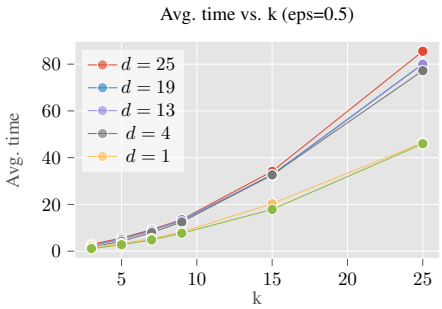


Figure 5: Average epoch time vs. k, circle loss, $\epsilon = 0.5$.