MELCOT: A Hybrid Learning Architecture with Marginal Preservation for Matrix-Valued Regression

Khang Tran
khangtranchicken2006@gmail.com
National University of Singapore
Singapore

Hieu Cao
caothanhhieu2013@gmail.com
University of Science - VNUHCM
Ho Chi Minh City, Vietnam

Thinh Pham
thinhcm2003@gmail.com
University of Science - VNUHCM
Ho Chi Minh City, Vietnam

Nghiem Diep
nghiemdt2013@gmail.com
University of Science - VNUHCM
Ho Chi Minh City, Vietnam

Tri Cao*
tricao2001vn@gmail.com
National University of Singapore
Singapore

Binh Nguyen*
ngtbinh@hcmus.edu.vn
University of Science - VNUHCM
Ho Chi Minh City, Vietnam

Abstract

Regression is essential across many domains but remains challenging in high-dimensional settings, where existing methods often lose spatial structure or demand heavy storage. In this work, we address the problem of matrix-valued regression, where each sample is naturally represented as a matrix. We propose MELCOT, a hybrid model that integrates a classical machine learning–based Marginal Estimation (ME) block with a deep learning–based Learnable-Cost Optimal Transport (LCOT) block. The ME block estimates data marginals to preserve spatial information, while the LCOT block learns complex global features. This design enables MELCOT to inherit the strengths of both classical and deep learning methods. Extensive experiments across diverse datasets and domains demonstrate that MELCOT consistently outperforms all baselines while remaining highly efficient.

CCS Concepts

• **Computing methodologies** → **Supervised learning by regression**; *Neural networks*; Support vector machines.

Keywords

Tensor Regression, Optimal Transport, Tabular Data

1 Introduction

Regression is a core task in machine learning for modeling relationships between input and output variables. Matrix-valued regression (MVR) extends this paradigm by representing both inputs and outputs as matrices, thereby preserving row–column dependencies. This formulation naturally arises in many real-world applications. For example, in international sports analytics, one can model the relationship between country-level factors such as GDP, population, and life expectancy and the resulting medal distributions across different sports, naturally represented as input and output matrices. Despite its importance, a key challenge in MVR lies in designing models that can effectively preserve spatial structure while remaining computationally efficient.

Existing approaches to high-dimensional regression can be divided into three main families. Classical machine learning models such as SVM [2], random forests [25], and boosting methods are computationally efficient and interpretable but require vectorization, which inevitably discards important row–column interactions and spatial dependencies [10, 27]. Tensor-based methods, including CP, Tucker, and multilinear PLS [11, 21, 35], together with their nonlinear or probabilistic extensions such as kernel ridge regression [22], Gaussian process regression [36], and random forest–based tensor regression [14], better preserve multiway structure but remain largely linear or low-rank and therefore limited in expressiveness for complex data. Deep learning approaches, ranging from general neural architectures [15, 19] to CNN-based tensor regression networks and low-rank extensions [4, 18], currently define the state of the art, yet they are parameter-heavy, computationally expensive, and often difficult to scale effectively in real-world scenarios.

In light of these challenges, Optimal Transport (OT) provides a promising direction for regression with structured data. Originally introduced by Monge and later formalized by Kantorovich [5, 16], OT offers a principled framework to align distributions while preserving structural information, and has been successfully applied in economics, imaging, and biology[3, 24, 28]. Cuturi's Sinkhorn algorithm [9] reduced the computational complexity of OT to practical levels, while recent advances that learn cost matrices directly from data [7, 29] have further enhanced its adaptability. These developments make OT particularly appealing for matrix-valued regression, where preserving local marginals while capturing global dependencies is essential.

In this work, we propose MELCOT, a hybrid architecture that integrates a machine learning–based Marginal Estimation (ME) block with a deep learning–based Learnable-Cost Optimal Transport (LCOT) block. The ME block preserves spatial information by estimating row and column marginals with lightweight computations, ensuring efficiency. The LCOT block leverages learnable OT to capture complex global dependencies, and can be parameterized with shallow neural networks to avoid excessive computational cost. This design allows MELCOT to balance efficiency with expressive power, overcoming the limitations of existing methods. Our contributions are three-fold: (1) We introduce MELCOT, a novel hybrid architecture that explicitly preserves marginal structure while modeling global dependencies through OT; (2) We conduct comprehensive experiments on the matrix-valued regression (MVR) problem, benchmarking MELCOT against classical, deep, and tensor-based baselines in both linear and nonlinear settings; and (3) We show that

*Correspoding authors

MELCOT consistently outperforms all baselines while maintaining high efficiency.

2 Preliminaries

Optimal Transport (OT). Created to provide a principled way to compare probability distributions by computing the most efficient plan to transform one distribution into another. Specifically, for a positive integer n, let $\mathbb{R}_+^n$ denote the space of n-dimensional vectors with non-negative real entries. Let $\mathbf{m}_1 \in \mathbb{R}_+^{n_1}, \mathbf{m}_2 \in \mathbb{R}_+^{n_2}$ be two discrete probability distributions (i.e., vectors whose entries are non-negative and sum to 1). We define the feasible set $\mathcal{T}$ as follows:

$$\mathcal{T}(\mathbf{m}_1, \mathbf{m}_2) = \{\mathbf{X} \in \mathbb{R}_+^{n_1 \times n_2} : \mathbf{X}\mathbf{1}_{n_2} = \mathbf{m}_1, \mathbf{X}^\top \mathbf{1}_{n_1} = \mathbf{m}_2\}, \quad (1)$$

where $\mathbf{1}_n$ is a $n-$dimension vector with each entry equal to 1. Consider a cost matrix $\mathbf{C} \in \mathbb{R}_+^{n_1 \times n_2}$, the Optimal Transport problem [32] can now be formally expressed as finding:

$$\mathrm{OT}(\mathbf{m}_1, \mathbf{m}_2, \mathbf{C}) = \min_{\mathbf{X} \in \mathcal{T}(\mathbf{m}_1, \mathbf{m}_2)} \langle \mathbf{C}, \mathbf{X} \rangle, \quad (2)$$

where $\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{Tr}(\mathbf{A}^\top \mathbf{B})$ is the Frobenius inner product. In this work, we employ the Sinkhorn algorithm [9] to compute the OT map. This solver incorporates an additional entropic regularization term, controlled by a parameter ε. Details on how this parameter is selected are provided in Section 4.

Learnable-Cost Optimal Transport. This method, also known as Inverse Optimal Transport [7, 29], replaces a fixed, deterministic cost function with one that is learned directly from data through training. More formally, let $f_\theta : \Omega \to \mathbb{R}^{n_1 \times n_2}$ denote the *cost function*, parameterized by θ, where Ω is the data space. Let $\mathbf{X}_{\hat{\theta}}$ be the solution to Equation 2, where the cost matrix $\mathbf{C}$ is computed via $f_{\hat{\theta}}$. The Learnable-Cost Optimal Transport problem aims to learn the optimal parameters θ by minimizing the discrepancy between the resulting transport plan and a ground-truth plan from training data:

$$\arg\min_\theta \mathcal{L}(\mathbf{X}_\theta, \mathbf{X}_{\text{ground truth}}),$$

where $\mathbf{X}_{\text{ground truth}}$ denotes the observed ground truth, and $\mathcal{L}$ is a predefined loss function measuring the alignment between the predicted value and the ground truth.

Matrix-Valued Regression (MVR). In the MVR problem, given a three-dimensional training dataset $\mathcal{D} = \{(\mathbf{M}_t, \hat{\mathbf{M}}_t)\}_{t=1}^T$, where each sample $\mathbf{M}_t$ is an $m \times n$ matrix and $\hat{\mathbf{M}}_t$ its corresponding $m' \times n'$ label. Our goal is to learn a model function $g : \mathbb{R}^{m \times n} \to \mathbb{R}^{m' \times n'}$ that minimizes a pre-chosen loss $\mathcal{L}\big(g(\mathbf{M}_t), \hat{\mathbf{M}}_t\big)$ over $\mathcal{D}$. At test time, for a new input $\mathbf{M}_{\text{test}}$, we compute

$$\mathbf{M}_{\text{pred}} = g(\mathbf{M}_{\text{test}}),$$

and evaluate performance of the model g by comparing $\mathbf{M}_{\text{pred}}$ to the true label via different given metrics. Notably, in the special case $n = 1$, each $\mathbf{M}_t$ collapses to a vector in $\mathbb{R}^m$, recovering the standard regression setting. To address the MVR problem, conventional machine learning models typically flatten the data and reformulate it as a standard vector regression task. Specifically, each matrix-valued pair $(\mathbf{M}_i, \hat{\mathbf{M}}_i) \in \mathbb{R}^{m \times n} \times \mathbb{R}^{m' \times n'}$ is flatten via $\mathbf{m}_i = \mathrm{vec}(\mathbf{M}_i) \in \mathbb{R}^{mn}$ and $\hat{\mathbf{m}}_i = \mathrm{vec}(\hat{\mathbf{M}}_i) \in \mathbb{R}^{m'n'}$. The resulting dataset $\mathcal{D}' = \{(\mathbf{m}_t, \hat{\mathbf{m}}_t)\}_{t=1}^T$ is used to train the model $g : \mathbb{R}^{mn} \to \mathbb{R}^{m'n'}$ as a standard regression problem. The prediction result $\hat{\mathbf{m}}_{\text{pred}} = g(\mathbf{m}_{\text{test}})$ is then reshaped back to matrix form using the inverse mapping $\mathrm{mat} : \mathbb{R}^{m'n'} \to \mathbb{R}^{m' \times n'}$. However, this approach leads to the loss of spatial information as a result of the flattening and concatenation process. Hence, more advanced techniques, which we have introduced in the introduction, need to be developed to adapt to matrix data.

3 Main Architecture

In this section, we describe the construction of our proposed architecture. We first present the design of each block, followed by the end-to-end training and testing procedure. The MELCOT framework comprises two main components: the ME block and the LCOT block, which are optimized independently during training and integrated at inference. An overview is provided in Figure 1.

3.1 ME Block

To ensure that our model preserves spatial information, we introduce the Marginal Estimation (ME) block. This block captures the row/column marginals of the target matrix so that spatial structure is retained and subsequently leveraged by the transport module in LCOT. In practice, some marginals may be known in advance (*e.g.*, Olympic medal counts; see Section 4). In such cases, the ME block predicts only the unknown marginals, so that $\mathcal{M}_r$, $\mathcal{M}_c$, or neither requires training.

Formally, ME Block comprises two vector regressors: $\mathcal{M}_c$ for column margins and $\mathcal{M}_r$ for row margins. Given a training pair $\mathbf{M}_i \in \mathbb{R}^{m \times n}$ and ground truth $\widehat{\mathbf{M}}_i \in \mathbb{R}^{m' \times n'}$, we flatten $\mathbf{M}_i$ into $\mathbf{m}_i \in \mathbb{R}^{mn}$ and pass it through $\mathcal{M}_c$ and $\mathcal{M}_r$ to produce predicted marginals $\mathbf{m}^c_{\text{pred}}$ and $\mathbf{m}^r_{\text{pred}}$. Ground-truth marginals are obtained by marginalizing $\widehat{\mathbf{M}}_i$, yielding $\mathbf{m}^c_{\text{GT}}$ and $\mathbf{m}^r_{\text{GT}}$. The ME losses are

$$\mathcal{L}_{\mathcal{M}_c}\big(\mathbf{m}^c_{\text{pred}}, \mathbf{m}^c_{\text{GT}}\big), \qquad \mathcal{L}_{\mathcal{M}_r}\big(\mathbf{m}^r_{\text{pred}}, \mathbf{m}^r_{\text{GT}}\big),$$

and in our implementation, we use mean-squared error (MSE) for both $\mathcal{L}_{\mathcal{M}_c}$ and $\mathcal{L}_{\mathcal{M}_r}$.

3.2 LCOT Block

The LCOT block reconstructs the full matrix by solving an optimal transport problem conditioned on the predicted marginals and a learned cost matrix. It consists of (i) a *cost function* that maps the input to a cost matrix and (ii) an *OT module* that solves the transport problem using these inputs.

Cost Function. Having fixed the marginals (either predicted by the ME block or known a priori), the LCOT block learns a data-dependent transport cost. Concretely, for each training pair we take the same input matrix $\mathbf{M}_i \in \mathbb{R}^{m \times n}$ and form its vectorization $\mathbf{m}_i = \mathrm{vec}(\mathbf{M}_i) \in \mathbb{R}^{mn}$. This vector is fed to a two-layer multilayer perceptron, and the network output is reshaped to match the cost-matrix dimensions used by the OT module:

$$\mathbf{m}_i \xrightarrow{\text{2-layer MLP}} \mathrm{vec}(\mathbf{C}_{\text{pred}}) \ \ \text{reshape} \ \ \to \ \ \mathbf{C}_{\text{pred}} \in \mathbb{R}^{m' \times n'}.$$

Intuitively, $\mathbf{m}_i$ summarizes the observed input structure, while the ME block supplies the target row/column totals; the cost network then maps $\mathbf{m}_i$ to a matrix $\mathbf{C}_{\text{pred}}$ that guides the OT plan consistent with those marginals. This cost function is the only learnable component in LCOT; it is trained jointly with the OT solver in the

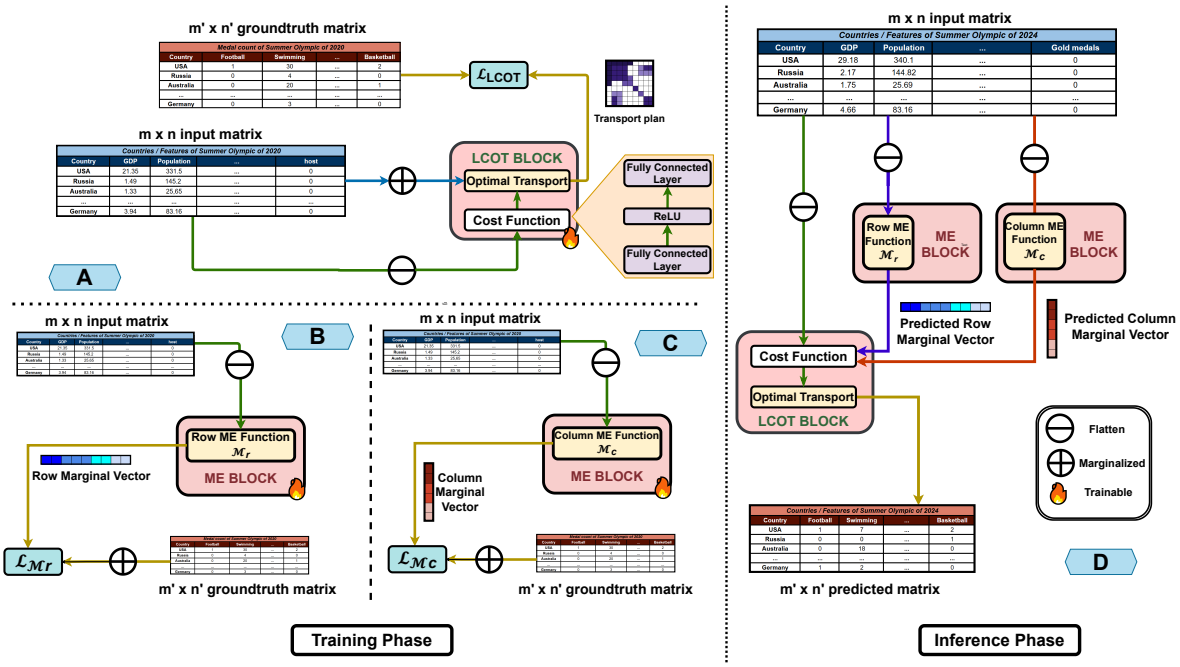

Figure 1: Overview of the MELCOT pipeline: (A) training phase of the LCOT block; (B) training phase of the ME row block $\mathcal{M}_r$; (C) training phase of the ME column block $\mathcal{M}_c$; and (D) inference phase of MELCOT. The pipeline begins with input data being used to train modules A, B, and C independently. During inference, blocks B and C are executed first to estimate the marginals, which, together with the original input data, are then passed into the trained LCOT block (A) to get the final prediction.

LCOT objective (see below), while the marginals used by OT come from ME (or prior knowledge) and are not backpropagated through this module.

Optimal Transport Module. As in Subsection 2, the module solves Eq. 2 given two marginals and the cost $\mathbf{C}_{\text{pred}}$, returning a plan $\mathbf{T}_i \in \mathbb{R}^{m' \times n'}$. During training, $\mathbf{m}_r, \mathbf{m}_c$ are the row/column sums of the ground truth $\widehat{\mathbf{M}}_i$; at inference, they are either known a priori (e.g., Olympic counts) or predicted by the ME block. The plan satisfies mass constraints $\mathbf{T}_i \mathbf{1}_{n'} = \mathbf{m}_r$, $\mathbf{T}_i^\top \mathbf{1}_{m'} = \mathbf{m}_c$ and minimizes transport cost under $\mathbf{C}_{\text{pred}}$. For efficiency and differentiability, we use the Sinkhorn algorithm [9] with entropic regularization ε and tolerance γ; ablations of OT solvers are in Table 2. Let $\mathbf{T}_i$ denote the transport plan obtained by solving Equation 2 with inputs $(\mathbf{m}_r, \mathbf{m}_c, \mathbf{C}_{\text{pred}})$. The LCOT block is trained by minimizing

$$\mathcal{L}_{\text{LCOT}}(\mathbf{T}_i, \widehat{\mathbf{M}}_i) = \|\mathbf{T}_i - \widehat{\mathbf{M}}_i\|_F.$$

3.3 Training and Testing Procedures

Training. We optimize the ME and LCOT blocks independently. For each training pair $(\mathbf{M}_i, \widehat{\mathbf{M}}_i)$: (i) ME: minimize $\mathcal{L}_{\mathcal{M}_c}$ and $\mathcal{L}_{\mathcal{M}_r}$ using inputs $\mathbf{m}_i = \text{vec}(\mathbf{M}_i)$ and targets from the marginals of $\widehat{\mathbf{M}}_i$; (ii) LCOT: predict $\mathbf{C}_{\text{pred}}$ from $\mathbf{m}_i$, solve Equation 2 with $(\mathbf{m}_r, \mathbf{m}_c, \mathbf{C}_{\text{pred}})$ (Sinkhorn with ε, γ), and minimize $\mathcal{L}_{\text{LCOT}}$.

Testing. Given a test sample $\mathbf{M}$, we form $\mathbf{m} = \text{vec}(\mathbf{M})$. If prior marginals are available, we use them; otherwise, we obtain $\mathbf{m}^r_{\text{pred}}$ and $\mathbf{m}^c_{\text{pred}}$ from the ME block. In parallel, the cost function maps $\mathbf{m}$ to $\mathbf{C}_{\text{pred}}$. The OT module then solves Equation 2 (via Sinkhorn) with these inputs to produce the transport plan $\mathbf{T}_{\text{pred}}$, which is compared to $\widehat{\mathbf{M}}$ under the evaluation metrics.

For MELCOT to function, solving Equation 2 is essential in both training and testing; therefore, both the LCOT and ME blocks are compulsory components of the model. However, smaller elements within these blocks, such as the OT solver or the cost function parameterization, can be varied without compromising the overall framework.

4 Experimental Results

Dataset. We evaluate our approach on three datasets. The Olympic Medal dataset (compiled from [33] for economic indicators, [23] for life expectancy, and [8] for historical medal counts) predicts medal distributions across sports, with an input–output mapping of $\mathbb{R}^{13\times84} \to \mathbb{R}^{26\times84}$. Notably, in the Olympic Medal dataset, the column marginals are known in advance. These correspond to the total number of medals awarded in each sport, which remain constant across different years. The Electricity Production dataset [30] models energy production from fossil fuels, nuclear, and renewable sources for 104 countries, with shape $\mathbb{R}^{13\times104} \to \mathbb{R}^{3\times104}$. The Tourism dataset [31] forecasts key tourism indicators (arrivals, departures, exports, receipts) across 31 countries, represented as tensors of $\mathbb{R}^{4\times31} \to \mathbb{R}^{4\times31}$.

Baselines. We compare MELCOT against four groups of baselines:

Model	Electricity Production			Olympic Medal			Tourism		
	rMSE↓	RSE↓	Time↓	rMSE↓	RSE↓	Time↓	rMSE↓	RSE↓	Time↓
XGBoost[6]	0.009	0.25	27.2	0.60	0.52	183.8	0.100	1.27	11.3
AdaBoost[12]	0.006	0.16	228.1	0.54	0.47	223.4	0.141	1.67	156.3
RF [25]	0.007	0.19	32.7	0.57	0.49	225.9	0.141	1.60	12.9
SVM [2]	0.010	0.32	10.8	0.53	0.46	86.2	0.100	1.38	5.2
DT [26]	0.010	0.18	9.9	0.57	0.50	67.8	0.300	1.07	4.2
DNN 1 layer	0.024	0.65	1.5	1.06	0.92	1.7	0.632	7.46	1.4
DNN 2 layer	0.032	0.86	1.5	1.05	0.92	1.8	0.316	4.23	1.4
TabNet [1]	0.020	0.58	1.6	0.72	0.63	1.5	0.141	1.10	1.2
FT-T [13]	0.020	0.45	43.1	0.54	0.48	28.7	0.055	0.66	1.6
CP [37]	0.013	0.36	0.2	0.78	0.47	0.5	0.055	0.62	1.68
Tucker [20]	0.004	0.12	1.3	0.77	0.46	8.29	0.082	0.95	2.1
PLS [34]	0.006	0.16	**0.04**	0.83	0.50	**0.08**	0.069	0.80	**0.06**
ResNetCP[4]	0.014	0.36	220.7	0.79	0.47	190.8	0.126	1.49	175.0
ResNetT[4]	0.020	0.54	326.9	0.77	0.46	233.3	0.130	1.50	189.9
ResNetTT [4]	0.017	0.48	192.5	0.75	0.46	184.4	0.133	1.53	173.4
$MELCOT_{SVM}$	0.004	0.12	5.9	0.52	**0.45**	7.0	**0.017**	**0.19**	2.5
$MELCOT_{RF}$	**0.003**	**0.05**	16.7	**0.51**	**0.45**	17.7	**0.017**	**0.19**	6.6

Table 1: Performance of MELCOT and baselines across datasets. $MELCOT_{SVM}$ and $MELCOT_{RF}$ denote MELCOT with the ME block implemented as SVM and RF, respectively. Models are evaluated on different metrics, including rMSE (root MSE), RSE, and inference time (ms). Best and second-best scores are highlighted in bold and underlined, respectively.

Method	Training Time↓	Inference Time↓	rMSE↓
EOT	0.03	**0.002**	**0.017**
EPOT	0.04	0.006	0.045
LPOT	**0.02**	0.005	0.055
LPPOT	**0.02**	0.005	0.055

Table 2: Ablation of different OT variants on the Tourism dataset within the OT module, evaluated by training time (seconds/iteration), inference time (seconds), and performance (rMSE). Best results are in bold.

(i) traditional machine learning models including XGBoost [6], AdaBoost [12], Random Forest (RF) [25], Support Vector Machine (SVM) [2], and Decision Tree (DT) [26]; (ii) deep learning models including 1-layer and 2-layer fully connected networks, TabNet [1], and Feature Tokenizer-Transformer (FT-T) [13]; (iii) tensor-based models including Canonical Polyadic (CP) [37], Tucker [20], and Partial Least Squares (PLS) [34]; and (iv) deep tensor-based models consisting of ResNet18 with a Tensor Regression Layer [18] and three low-rank tensor decomposition variants, CP, Tucker, and Tensor Train (TT) following [4], referred to in this paper as ResNetCP, ResNetT, and ResNetTT.

Parameters and Fine-tuning. For ME Block, we employ two approaches: SVM and Random Forest, each applied to both $\mathcal{M}_r$ and $\mathcal{M}_c$. For the LCOT block, the cost function is optimized with ADAM solver [17] using a learning rate of 1×10^{-2}. The cost model is a 2-layer DNN with input and output dimensions of 10; the hidden dimension is dataset-specific: 30 for the Electricity Production dataset and 10 for the others. For OT module, we set the regularization coefficient to $\varepsilon = 5{\times}10^{-1}$, the convergence tolerance to $\gamma = 5{\times}10^{-5}$, and the maximum iterations to 1000.

Results. We use rMSE and RSE for evaluation metrics and compare inference time between baselines and our methods. The results, summarized in Table 1, demonstrate that our proposed method consistently outperforms all baselines across the three datasets. On the *Electricity Production* dataset, MELCOT achieves the lowest rMSE (0.003) and RSE (0.05), improving upon strong baselines such as PLS and Tucker. For the *Olympic Medal* dataset, MELCOT again obtains the best results with rMSE of 0.51 and RSE of 0.45, outperforming tree-based models (AdaBoost) and tensor methods (CP). On the *Tourism* dataset, MELCOT significantly surpasses baselines with an rMSE of 0.017 and RSE of 0.19, while most baselines yield RSE > 1. With respect to inference time, our method is only slower than small neural networks, whose simplicity makes them computationally lightweight, and traditional tensor methods, which reduce complexity by projecting data into low-dimensional representations. Nevertheless, our method consistently outperforms these approaches in terms of accuracy across the evaluated metrics.

Ablation on OT variants. In previous sections, experiments were conducted using OT with the Sinkhorn solver. Nevertheless, OT admits several variants, among which is Partial Optimal Transport (POT). The key idea of POT is to transport only a portion of the total mass rather than the entirety. Formally, consider the same setting as in Subsection 2, and define the feasible set of POT, denoted by $\mathcal{R}$, as follows:

$$\mathcal{R}(\mathbf{m}_1,\mathbf{m}_2,s) = \{\mathbf{X} \in \mathbb{R}_+^{n_1\times n_2} : \mathbf{X}\mathbf{1}_{n_2} \le \mathbf{m}_1, \mathbf{X}^\top \mathbf{1}_{n_1} \le \mathbf{m}_2, \mathbf{1}_{n_1}^\top \mathbf{X}\mathbf{1}_{n_2} = s\},$$

where s is the transport mass. Consider a cost matrix $\mathbf{C} \in \mathbb{R}_+^{n_1\times n_2}$, the POT objective can now be written as:

$$\mathrm{POT}(\mathbf{m}_1, \mathbf{m}_2, \mathbf{C}, s) = \min_{\mathbf{X}\in\mathcal{R}(\mathbf{m}_1,\mathbf{m}_2,s)} \langle \mathbf{C}, \mathbf{X}\rangle, \tag{3}$$

Furthermore, besides the Sinkhorn algorithm, both OT and POT can also be solved via Linear Programming (LP). For clarity, we denote Sinkhorn-based methods as EOT and EPOT, and LP-based methods as LPOT and LPPOT. We evaluate these four variants on the Tourism dataset using SVM for both $\mathcal{M}_r$ and $\mathcal{M}_c$, comparing training time, inference time, and performance (Table 2). Overall, EOT, though slightly slower in training, achieves superior performance and inference efficiency compared to the other variants.

Implementation Details. All experiments are carried out on a system with an 8-core CPU and 32 GB RAM.

5 Conclusion

In this work, we propose MELCOT, a hybrid architecture that combines a deep learning–based LCOT block with a machine learning–based ME block for matrix-valued regression. By estimating marginals independently and reconstructing the output via OT, MELCOT preserves spatial structure in the data. Experiments show that MELCOT consistently outperforms traditional deep learning, machine learning, and tensor-based methods across datasets of varying shapes, sizes, and noise levels. For future work, one can extend MELCOT to higher-dimensional settings or evaluate its performance in stochastic scenarios where data arrives sequentially.

6 GenAI Usage Disclosure

In this work, we used ChatGPT and Grammarly to check for grammar errors and enhance clarity and coherence. While generative AI tools were employed during the writing process, the authors retain full responsibility for the content.

References

[1] Sercan Ö Arik and Tomas Pfister. 2021. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 6679–6687.

[2] Bernhard Boser, Isabelle Guyon, and Vladimir Vapnik. 1996. A Training Algorithm for Optimal Margin Classifier. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory* 5 (08 1996). doi:10.1145/130385.130401

[3] Olivier Bousquet, Sylvain Gelly, Ilya Tolstikhin, Carl-Johann Simon-Gabriel, and Bernhard Schoelkopf. 2017. From optimal transport to generative modeling: the VEGAN cookbook. *arXiv preprint arXiv:1705.07642* (2017).

[4] Xingwei Cao and Guillaume Rabusseau. 2017. Tensor regression networks with various low-rank tensor approximations. *arXiv preprint arXiv:1712.09520* (2017).

[5] Cayley. 1882. On Monge's "Mémoire sur la Théorie des Déblais et des Remblais.". *Proceedings of the London Mathematical Society* s1-14, 1 (11 1882), 139–143. doi:10.1112/plms/s1-14.1.139 arXiv:https://academic.oup.com/plms/article-pdf/s1-14/1/139/4346565/s1-14-1-139.pdf

[6] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2* 1, 4 (2015), 1–4.

[7] Wei-Ting Chiu, Pei Wang, and Patrick Shafto. 2022. Discrete probabilistic inverse optimal transport. In *International Conference on Machine Learning*. PMLR, 3925–3946.

[8] Consortium for Mathematics and Its Applications (COMAP). 2025. *2025 MCM Problem C Data Archive.* https://www.immchallenge.org/mcm/2025_Problem_C_Data.zip Includes population data from 1970 to 2022 for Problem C: "Models for Olympic Medal Tables".

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

[10] Tri Doan and Jugal Kalita. 2015. Selecting machine learning algorithms using regression models. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 1498–1505.

[11] Nicolaas Klaas M Faber, Rasmus Bro, and Philip K Hopke. 2003. Recent developments in CANDECOMP/PARAFAC algorithms: a critical review. *Chemometrics and Intelligent Laboratory Systems* 65, 1 (2003), 119–137.

[12] Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55, 1 (1997), 119–139.

[13] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. *Advances in neural information processing systems* 34 (2021), 18932–18943.

[14] Huyan Huang, Yipeng Liu, Zhen Long, and Ce Zhu. 2020. Robust low-rank tensor ring completion. *IEEE Transactions on Computational Imaging* 6 (2020), 1117–1126.

[15] Sadiq Hussain, Silvia Gaftandzhieva, Md Maniruzzaman, Rositsa Doneva, and Zahraa Fadhil Muhsin. 2021. Regression analysis of student academic performance using deep learning. *Education and Information Technologies* 26, 1 (2021), 783–798.

[16] L. Kantorovich. 1942. On the transfer of masses (in Russian). 227 pages. https://cir.nii.ac.jp/crid/1370565168575910170

[17] Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Jean Kossaifi, Zachary C Lipton, Arinbjorn Kolbeinsson, Aran Khanna, Tommaso Furlanello, and Anima Anandkumar. 2020. Tensor regression networks. *Journal of Machine Learning Research* 21, 123 (2020), 1–21.

[19] Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda-Pineda, and Radu Horaud. 2019. A comprehensive analysis of deep regression. *IEEE transactions on pattern analysis and machine intelligence* 42, 9 (2019), 2065–2081.

[20] Lexin Li and Xin Zhang. 2017. Parsimonious tensor response regression. *J. Amer. Statist. Assoc.* 112, 519 (2017), 1131–1146.

[21] Xiaoshan Li, Da Xu, Hua Zhou, and Lexin Li. 2018. Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences* 10, 3 (2018), 520–545.

[22] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, Vol. 2. IEEE, 416–423.

[23] Our World in Data. 2024. *Life Expectancy vs Health Expenditure.* https://ourworldindata.org/grapher/life-expectancy-vs-health-expenditure Accessed: 2025-05-24.

[24] Julien Rabin and Nicolas Papadakis. 2015. Non-convex relaxation of optimal transport for color transfer between images. In *Geometric Science of Information: Second International Conference, GSI 2015, Palaiseau, France, October 28-30, 2015, Proceedings 2*. Springer, 87–95.

[25] Steven J Rigatti. 2017. Random forest. *Journal of Insurance Medicine* 47, 1 (2017), 31–39.

[26] Lior Rokach and Oded Maimon. 2005. Decision trees. *Data mining and knowledge discovery handbook* (2005), 165–192.

[27] Shen Rong and Zhang Bao-Wen. 2018. The research of regression model in machine learning field. In *MATEC Web of Conferences*, Vol. 176. EDP Sciences, 01033.

[28] Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. 2019. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell* 176, 4 (2019), 928–943.

[29] Andrew M Stuart and Marie-Therese Wolfram. 2020. Inverse optimal transport. *SIAM J. Appl. Math.* 80, 1 (2020), 599–619.

[30] Ansh Tanwar. 2023. *Global Data on Sustainable Energy (2000–2020).* https://www.kaggle.com/datasets/anshtanwar/global-data-on-sustainable-energy/data Accessed: 2025-05-24.

[31] Bushraq Urban. n.d.. Tourism and Economic Impact. https://www.kaggle.com/datasets/bushraqurban/tourism-and-economic-impact. Accessed: 2025-05-24.

[32] Cédric Villani et al. 2008. *Optimal transport: old and new.* Vol. 338. Springer.

[33] World Bank. 2025. *World Bank Open Data.* https://data.worldbank.org/ Accessed: 2025-05-24.

[34] Qibin Z hao, Cesar F Caiafa, Danilo Mandic, Liqing Zhang, Tonio Ball, Andreas Schulze-Bonhage, and Andrzej Cichocki. 2011. Multilinear subspace regression: An orthogonal tensor decomposition approach. *Advances in neural information processing systems* 24 (2011).

[35] Xin Zhang and Lexin Li. 2017. Tensor envelope partial least-squares regression. *Technometrics* 59, 4 (2017), 426–436.

[36] Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. 2015. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE transactions on pattern analysis and machine intelligence* 37, 9 (2015), 1751–1763.

[37] Hua Zhou, Lexin Li, and Hongtu Zhu. 2013. Tensor regression with applications in neuroimaging data analysis. *J. Amer. Statist. Assoc.* 108, 502 (2013), 540–552.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009