OT4P: Unlocking Effective Orthogonal Group Path for Permutation Relaxation

Yaming Guo[1,4], Chen Zhu[2], Hengshu Zhu[3,4], Tieru Wu[1]

[1]Jilin University, [2]University of Science and Technology of China, [3]Chinese Academy of Sciences, [4]The Hong Kong University of Science and Technology (GZ)

Introduction

- **Problem.** Optimization over permutations is typically an NP-hard problem that arises extensively in ranking, matching, tracking, etc. Denoting the set of all n-order permutation matrices as $\mathcal{P}_n := \{P \in \{0,1\}^{n\times n} \mid \sum_i P_{i,j} = 1, \sum_j P_{i,j} = 1\ (\forall i,j)\}$, and this work considers optimization over permutation matrices:

$$\min_{P\in\mathcal{P}_n} f(P).$$

- **Relaxation methods.** Previous studies proposed relaxing permutation matrices into continuous spaces, including the convex hull of permutation matrices—the Birkhoff polytope—and their embeddings in a differentiable manifold—the orthogonal group. Recently, relaxation methods involving the Birkhoff polytope have made significant advancements, particularly in penalty-free optimization and probabilistic inference.
- **Motivation. However, providing equally good relaxation methods within the orthogonal group remains an unexplored area.** Indeed, relaxation onto the orthogonal group offers several unique potential advantages, such as:
  - lower representation dimension ($\frac{n(n-1)}{2}$) compared to Birkhoff polytope ($(n-1)^2$).
  - preserve the inner product of vectors, which maintain the geometric structures.

This work aims to **develop an effective method for relaxing the permutation matrices onto the orthogonal group**, with a particular focus on:

- Flexibility: can control the degree of approximation to permutation matrices.
- Simplicity: does not rely on additional penalty terms.
- Scalability: enables learning the latent variable model with permutations.

Method: OT4P

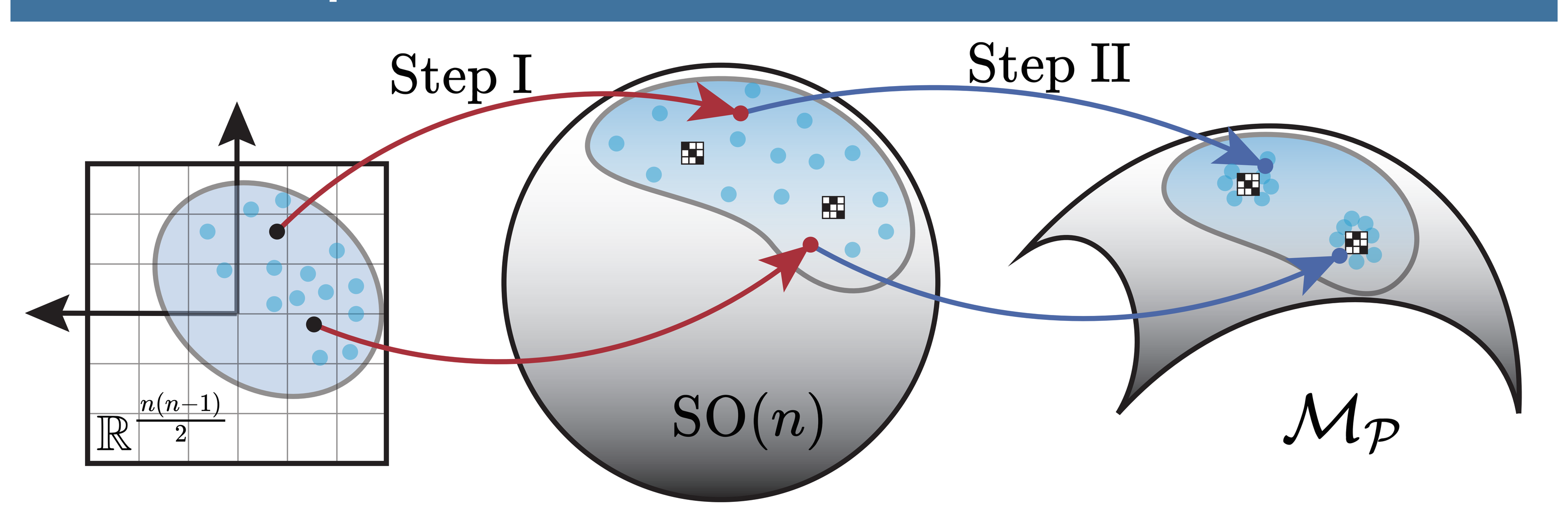

Figure 1. Illustration of OT4P with colored dots to help visualize the transformation.

This paper presents ***O**rthogonal Group-based **T**ransformation **for P**ermutation Relaxation* (OT4P), a temperature-controlled differentiable transformation. OT4P maps unconstrained vector space to the orthogonal group, where the temperature, in the limit, **concentrates orthogonal matrices near permutation matrices**.

As illustrated in Figure 1, OT4P involves two steps:

► **Step I** map a vector (●) to an orthogonal matrix (●) utilizing the Lie exponential:

$$\begin{aligned}\phi : \mathbb{R}^{\frac{n(n-1)}{2}} &\to \mathfrak{so}(n) &&\to \mathrm{SO}(n)\\ A &\mapsto A - A^\top &&\mapsto \mathrm{expm}(A - A^\top).\end{aligned} \tag{1}$$

► **Step II** move the orthogonal matrix (●) along the geodesic, controlled by temperature, to another orthogonal matrix (●), making it nearer to the closest permutation matrix (▚ or ▞):

$$\begin{aligned}\psi_\tau : \mathrm{SO}(n) &\to \mathcal{M}_\mathcal{P}\\ O &\mapsto \rho(O)D\left([\rho(O)D]^\top O\right)^\tau D^\top.\end{aligned} \tag{2}$$

Details of Step I

Detailed Equation (1)

- map a vector $A \in \mathbb{R}^{\frac{n(n-1)}{2}}$ to a skew-symmetric matrix $A - A^\top \in \mathfrak{so}(n)$.
- map a skew-symmetric matrix $A - A^\top \in \mathfrak{so}(n)$ to an orthogonal matrix $\mathrm{expm}(A - A^\top) \in \mathrm{SO}(n)$.

► The following theorem indicates that each orthogonal matrix in $\mathrm{SO}(n)$ can be represented by a vector in $\mathbb{R}^{\frac{n(n-1)}{2}}$, with each representation being uniquely defined within set $\mathcal{U}$, provided it exists there.

Theorem 1 in the paper

The mapping $\phi(\cdot)$ is differentiable, surjective, and it is injective on the domain $\mathcal{U} := \{A \in \mathbb{R}^{\frac{n(n-1)}{2}} \mid \mathrm{Im}\,\lambda_k(A - A^\top) \in (-\pi, \pi),\ \forall k\}$ with $\lambda_k(\cdot)$ the eigenvalues. Additionally, the set $\mathrm{SO}(n) \setminus \phi(\mathcal{U})$ has a zero Lebesgue measure in $\mathrm{SO}(n)$.

Boundary issues

The permutation matrices may include -1 as one of their eigenvalues, with their corresponding representations precisely lying on the boundary of $\mathcal{U}$. To avoid the optimization path to deviate from $\mathcal{U}$, we propose **shifting the boundary of $\mathcal{U}$ to other eigenvalues** by left-multiplying the result of $\phi(\cdot)$ with an orthogonal matrix $B \in \mathrm{SO}(n)$. Therefore, the representation of the permutation matrix P in $\mathcal{U}$ is changed from $\mathrm{logm}(P)$ to $\mathrm{logm}(B^\top P)$.

Details of Step II

Detailed Equation (2)

- find the permutation matrix $\rho(O) := \arg\max_{P\in\mathcal{P}_n} \langle P, O\rangle_\mathrm{F}$ closest to O.
- map P and O to the tangent space $T_P\mathrm{SO}(n)$ for linear interpolation, and then map the interpolation result back to $\mathrm{SO}(n)$, given as

$$\begin{aligned}\widetilde{O} &= P\,\mathrm{expm}(P^\top \left[\tau P\,\mathrm{logm}(P^\top O) + (1-\tau)P\,\mathrm{logm}(P^\top P)\right])\\ &= P(P^\top O)^\tau.\end{aligned} \tag{3}$$

► Equation (3) works only for even permutations; however, we can readily **extend it to the odd permutation** cases.

Extend to odd permutations

- identify an agent $\widehat{P} = PD$ of odd permutation P, with $D = \mathrm{diag}(\{1, \ldots, 1, -1\})$.
- move O toward $\widehat{P}$ to obtain $\widehat{O}$ using Equation (3).
- map $\widehat{O}$ to the neighborhood of P, resulting in $\widetilde{O} = \widehat{O}D^\top$.

► The following theorem shows that any point in the relaxation manifold $\mathcal{M}_\mathcal{P}$ of permutation matrices can be uniquely identified by an orthogonal matrix in the special orthogonal group $\mathrm{SO}(n)$, where the set of meaningless elements (i.e., not mapped any point in $\mathcal{M}_\mathcal{P}$) can be disregarded.

Theorem 2 in the paper

The mapping $\psi_\tau(\cdot)$ is differentiable, surjective, and injective on each submanifold $\mathcal{S}_P$. Additionally, the set of meaningless points for $\psi_\tau(\cdot)$ has a zero Lebesgue measure in $\mathrm{SO}(n)$.

Parameterization for gradient-based optimization

$$\min_{P\in\mathcal{P}_n} f(P) \xrightarrow{\text{relaxing}} \min_{A\in\mathbb{R}^{\frac{n(n-1)}{2}}} f(\psi_\tau \circ \phi(A)).$$

- ► The surjectivity does not alter the original problem.
- ► The injectivity does not complicate the original problem.
- ► The **efficient optimization process**.
  - Forward process. The orthogonal matrix O can be factorized as $O = Q\mathrm{diag}(\{\lambda_1, \ldots, \lambda_n\})Q^{-1}$, and then the matrix power O^τ can be computed by $O^\tau = Q\mathrm{diag}(\{\lambda_1^\tau, \ldots, \lambda_n^\tau\})Q^{-1}$.
  - Backward process. Given $\widetilde{O} = \psi_\tau(O)$, there exists a unique orthogonal matrix $W_\tau = \widetilde{O}O^\top$ such that $\widetilde{O} = W_\tau O$. In this way, the forward pass is streamlined into $\widetilde{O} = W_\tau O$, thereby rendering the backward pass highly efficient, as it only involves one linear transformation.

Re-parameterization provides stochastic optimization

$$\min \mathbb{E}_{P\sim q(P;\theta)} f(P).$$

- simulate $q(P;\theta)$ using the mappings $\rho(\cdot)$ and $\phi(\cdot)$:

$$P \sim q(P;\theta) \iff P = \rho(\phi(A + B\epsilon)) \text{ with } \theta := \{A, B \in \mathbb{R}^{\frac{n(n-1)}{2}}\}.$$

- **bring the gradient inside the expectation** by relaxing the mapping $\rho(\cdot)$ to $\psi_\tau(\cdot)$:

$$\nabla\mathbb{E}_{\epsilon\sim q(\epsilon)} f\left(\psi_\tau(\phi(A + B\epsilon))\right) = \mathbb{E}_{\epsilon\sim q(\epsilon)} \nabla f\left(\psi_\tau(\phi(A + B\epsilon))\right).$$

Experiments

► Finding mode connectivity

Table 1. ℓ_1-Distance and Precision (%) of algorithms across different network architectures.

Algorithm	MLP5		VGG11		ResNet18	
	$\log(1+\ell_1)$ (↓)	Precision (↑)	$\log(1+\ell_1)$ (↓)	Precision (↑)	$\log(1+\ell_1)$ (↓)	Precision (↑)
Weight Matching	0.000 ±0.00	100.0 ±0.00	0.000 ±0.00	100.0 ±0.00	1.215 ±2.72	99.97 ±0.06
Sinkhorn	0.000 ±0.00	100.0 ±0.00	11.61 ±0.07	63.08 ±3.14	9.830 ±0.181	95.56 ±0.88
OT4P ($\tau = 0.3$)	0.000 ±0.00	100.0 ±0.00	0.000 ±0.00	100.0 ±0.00	0.000 ±0.00	100.0 ±0.00
OT4P ($\tau = 0.5$)	0.000 ±0.00	100.0 ±0.00	0.818 ±1.83	99.99 ±0.03	0.000 ±0.00	100.0 ±0.00
OT4P ($\tau = 0.7$)	0.000 ±0.00	100.0 ±0.00	0.000 ±0.00	100.0 ±0.00	0.000 ±0.00	100.0 ±0.00

► Inferring neuron identities

Table 2. Marginal log-likelihood and Precision (%) of algorithms across different proportions of known neurons.

Algorithm	Known 5%		Known 10%		Known 20%	
	$\mathbb{E}\log p(Y\|P)$ (↑)	Precision (↑)	$\mathbb{E}\log p(Y\|P)$ (↑)	Precision (↑)	$\mathbb{E}\log p(Y\|P)$ (↑)	Precision (↑)
Naive	−3040 ±43.4	8.960 ±7.85	−2917 ±225	29.68 ±17.2	−1690 ±539	78.40 ±12.6
Gumbel-Sinkhorn	−2256 ±574	62.08 ±16.0	−239.8 ±119	98.16 ±1.95	−144.8 ±27.1	99.84 ±0.358
OT4P ($\tau = 0.3$)	−130.9 ±10.9	100.0 ±0.00	−127.5 ±10.1	100.0 ±0.00	−126.7 ±11.0	100.0 ±0.00
OT4P ($\tau = 0.5$)	−164.0 ±36.8	100.0 ±0.00	−149.7 ±25.0	100.0 ±0.00	−148.2 ±27.6	100.0 ±0.00
OT4P ($\tau = 0.7$)	−829.3 ±831	74.16 ±35.9	−183.1 ±46.2	100.0 ±0.00	−171.8 ±40.3	100.0 ±0.00

► Solving permutation synchronization

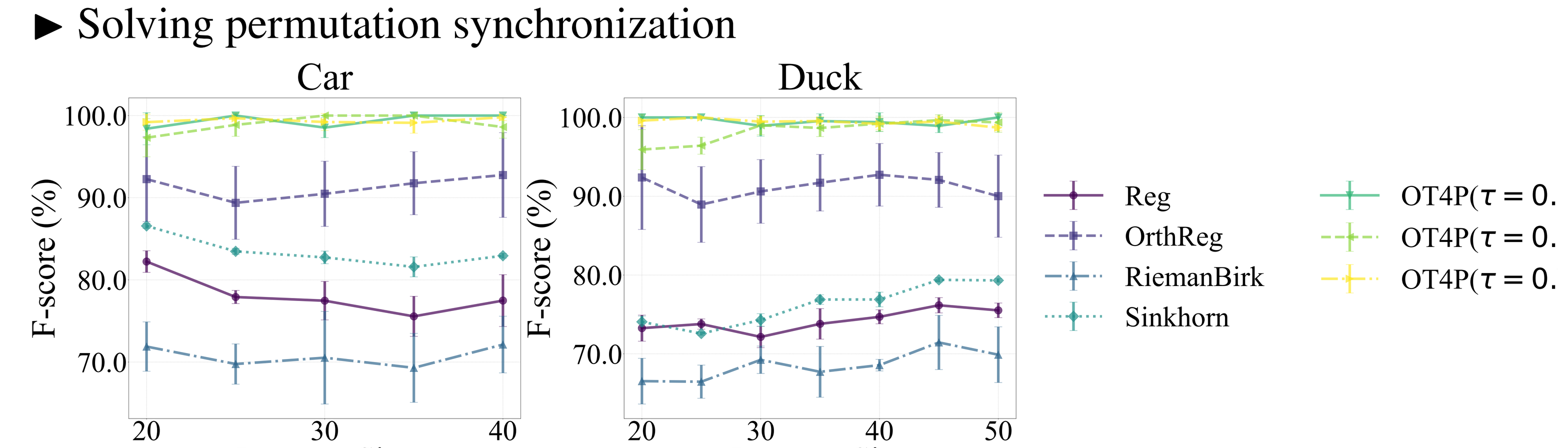

Figure 2. F-scores (%) for different algorithms on the WILLOW-ObjectClass dataset.