Preference Optimization for Combinatorial Optimization Problems

ICML International Conference On Machine Learning

Paper Available!

Mingjun Pan[1,2], Guanquan Lin*[3], You-wei Luo†[4], Bin Zhu[2], Zhien Dai, Lijun Sun[1], Chun Yuan†[3]*

[1]China Mobile [2] Peking University [3]Tsinghua University [4]Sun Yat-Sen University

mingjunpan96@gmail.com

TL;DR: We theoretically transform numerical rewards in RL4CO into pairwise preference signals and integrate local search during fine-tuning, empirically enabling faster convergence and higher-quality solutions for COPs like TSP, CVRP, and scheduling.

Background

- COPs are fundamental to practical applications like routing, circuit design, scheduling, and bioinformatics, but their NP-hard complexity prevents exact solution computation.
- Neural solvers provide efficient near-optimal solutions for large-scale COPs through two main approaches: Supervised Learning (SL) and Reinforcement Learning (RL).
- SL approaches require high-quality solution datasets for training.
- We **focus on RL**, which enables neural solvers to improve via trial-and-error ***without requiring pre-collected solutions***.

Key Challenges

- **Diminishing learning signals:** As the policy improves, the magnitude of advantage value decreases significantly. Since RL rely on these numerical signals to drive learning, the reduction in their scale leads to vanishing gradients and slow convergence.

$$\nabla_\theta J(\theta) = \mathbb{E}_{x\sim\mathcal{D},\tau\sim\pi_\theta(\tau|x)}\left[(r(x,\tau)-b(x))\nabla_\theta \log \pi_\theta(\tau \mid x)\right]$$

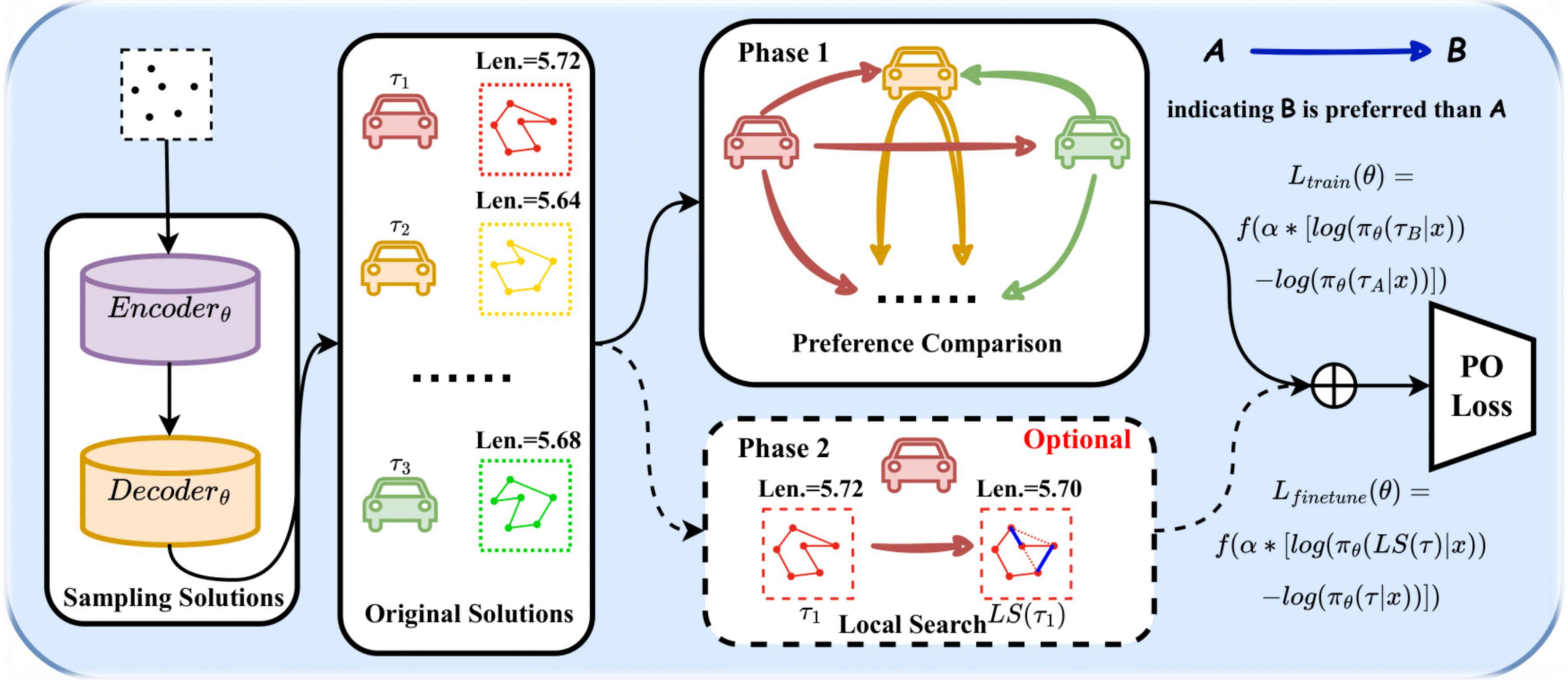
Decreases significantly

- **Unconstrained action spaces:** The vast combinatorial action spaces complicate efficient exploration, rendering exploration techniques like entropy regularization of trajectories computationally infeasible.

Computationally infeasible ⟹ $\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{\tau\sim\pi_\theta(\cdot|x)}\left[r(x,\tau)\right]+\alpha\mathcal{H}\left(\pi_\theta(\cdot \mid x)\right)\right]$

- **Additional inference time:** While neural solvers are efficient in inference, many works adopt techniques like local search as a post-processing step to further improve generated solutions, but incurs additional inference costs.

Methodology



Algorithmic Framework

- Starting from max-entropy RL

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{\tau\sim\pi_\theta(\cdot|x)}\left[r(x,\tau)\right]+\alpha\mathcal{H}\left(\pi_\theta(\cdot \mid x)\right)\right]$$

- Closed form of π and re-parameterized the reward function $\hat{r}$

$$\pi^*(\tau \mid x) = \frac{1}{Z(x)}\exp\left(\alpha^{-1} r(x,\tau)\right)$$

$$\hat{r}(x,\tau) = \alpha \log \pi(\tau \mid x) + \alpha \log Z(x)$$

- Preference-based RL Modeling

$$p^*(\tau_1 \succ \tau_2) = f\left(\hat{r}(x,\tau_1) - \hat{r}(x,\tau_2)\right)$$

$$p(\tau_1 \succ \tau_2 \mid x) = f\left(\alpha\left[\log \pi(\tau_1 \mid x) - \log \pi(\tau_2 \mid x)\right]\right)$$

- Training Objectives

$$\max_{\theta} \mathop{\mathbb{E}}_{x\sim\mathcal{D},\tau\sim\pi_\theta(\cdot|x)}\left[\mathbb{1}\left((r(x,\tau_1) > r(x,\tau_2)\right)\cdot \log p_\theta(\tau_1 \succ \tau_2 \mid x)\right]$$

Algorithm 1 Preference Optimization for COPs.

Input: problem set $\mathcal{D}$, number of training steps T, fine-tune steps $T_{\mathrm{FT}} \geq 0$,batch size B, learning rate η, ground truth reward function r, number of local search iteration I_{LS}, initialized policy π_θ.

for $step = 1,\ldots,T+T_{\mathrm{FT}}$ **do**

  //Sampling N solutions for each instance x_i

  $x_i \leftarrow \mathcal{D}, \quad \forall i \in \{1,\ldots,B\}$

  $\tau_i = \{\tau_i^1,\tau_i^2,\ldots,\tau_i^N\} \leftarrow \pi_\theta(x_i),\ \forall i \in \{1,\ldots,B\}$

  // Fine-tuning with LS for T_{FT} steps **(Optional)**

  if $step > T$ **then**

    $\{\hat{\tau}_i^1,\hat{\tau}_i^2,\ldots,\hat{\tau}_i^N\} \leftarrow$ *LocalSearch*$(\tau_i, r, I_{\mathrm{LS}}),\ \forall i$

    $\tau_i \leftarrow \tau_i \cup \{\hat{\tau}_i^1,\hat{\tau}_i^2,\ldots,\hat{\tau}_i^N\}$

  end if

  //Calculate conflict-free preference labels via grounding reward function $r(x,\tau)$

  $y_{j,k}^i \leftarrow \mathbb{1}\left(r(x_i,\tau_i^j) > r(x_i,\tau_i^k)\right),\ \forall j,k$

  //Approximating the gradient according to Eq. 8

  $\nabla_\theta J(\theta) \leftarrow \frac{\alpha}{B|\tau_i|^2}\sum_{i=1}^{B}\sum_{j=1,k=1}^{|\tau_i|}\left[\left(g(\tau_i^j,\tau_i^k,x_i) - g(\tau_i^k,\tau_i^j,x_i)\right)\nabla_\theta \log \pi_\theta(\tau_i^j \mid x_i)\right]$

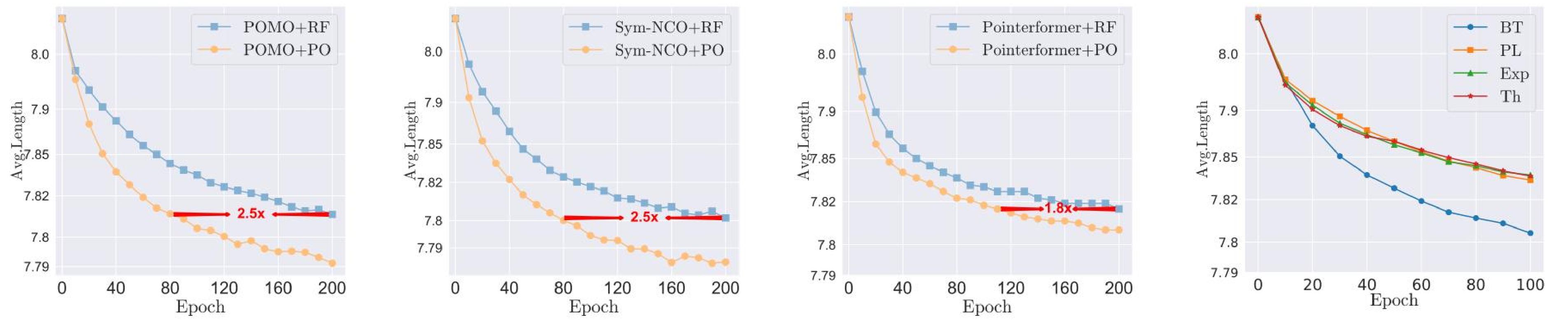
  $\theta \leftarrow \theta + \eta\nabla_\theta J(\theta)$

end for

Experiments

Comparison with Existing Algorithms on Standard Benchmarks

- Efficiency on convergence



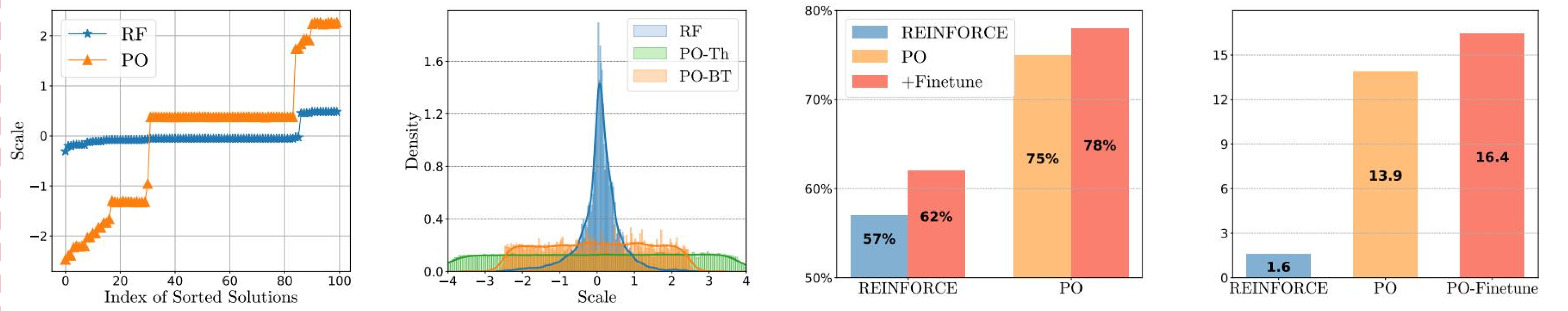
(a) POMO (b) Sym-NCO (c) Pointerformer (d) Preference Models

- Performance

Table 1: Experiment results on TSP and CVRP. Gap is evaluated on 10k instances and Times are summation of them.

	Solver	Algorithm	TSP						CVRP					
			$N=20$		$N=50$		$N=100$		$N=20$		$N=50$		$N=100$	
			Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
Heuristic	Concorde	-	0.00%	13m	0.00%	21.5m	0.00%	1.2h	-	-	-	-	-	-
	LKH3	-	0.00%	28s	0.00%	4.3m	0.00%	15.6m	0.09%	0.5h	0.18%	2h	0.55%	4h
	HGS	-	-	-	-	-	-	-	0.00%	1h	0.00%	3h	0.00%	5h
Neural Solvers	AM (Kool et al., 2019)	RF	0.28%	0.1s	1.66%	1s	3.40%	2s	4.40%	0.1s	6.02%	1s	7.69%	3s
		PO	0.33%	0.1s	1.56%	1s	2.86%	2s	4.60%	0.1s	5.65%	1s	6.82%	3s
	Pointerformer (Jin et al., 2023)	RF	**0.00%**	6s	0.02%	12s	0.15%	1m	-	-	-	-	-	-
		PO	**0.00%**	6s	0.01%	12s	0.06%	1m	-	-	-	-	-	-
	Sym-NCO (Kim et al., 2022)	RF	0.01%	1s	0.16%	2s	0.39%	8s	0.72%	1s	1.31%	4s	2.07%	16s
		PO	**0.00%**	1s	0.08%	2s	0.28%	8s	0.63%	1s	1.20%	4s	1.88%	16s
	POMO (Kwon et al., 2020)	RF	0.01%	1s	0.04%	15s	0.15%	1m	0.37%	1s	0.94%	5s	1.76%	3.3m
		PO	**0.00%**	1s	0.02%	15s	0.07%	1m	0.16%	1s	0.68%	5s	1.37%	3.3m
		PO+Finetune	**0.00%**	1s	**0.00%**	15s	**0.03%**	1m	**0.08%**	1s	**0.53%**	5s	**1.19%**	3.3m

How Effectively does PO Balance Exploitation and Exploration?



(a) Advantage Separation (b) Advantage Scales (c) Consistency of Policies (d) Trajectory Entropy

Distinction from RLHF

Our work distinguishes itself from preference optimization methods in RLHF especially for LLMs in a critical dimension. While RLHF typically relies on **subjective, offline** human-annotated datasets, our Preference Optimization framework for COPs employs an **active, online** learning strategy grounded in objective metrics (e.g., route length) to identify and prioritize superior solutions.