NEURAL INFORMATION PROCESSING SYSTEMS

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Learning Optimal Transport Plans Via Autoregressive Token Regression

Ivan Jaen Marquez, Sotirios Panagiotis Chytas, Vikas Singh

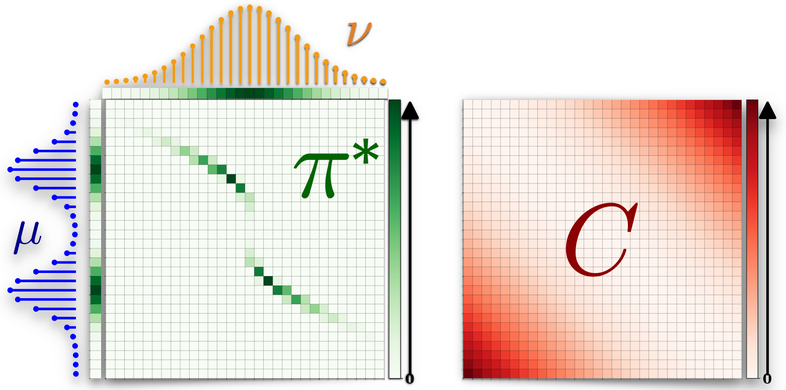
University of Wisconsin–Madison

go.wisc.edu/torot

40th Conference on Neural Information Processing Systems (NeurIPS 2026)

Discrete Optimal Transport (OT)

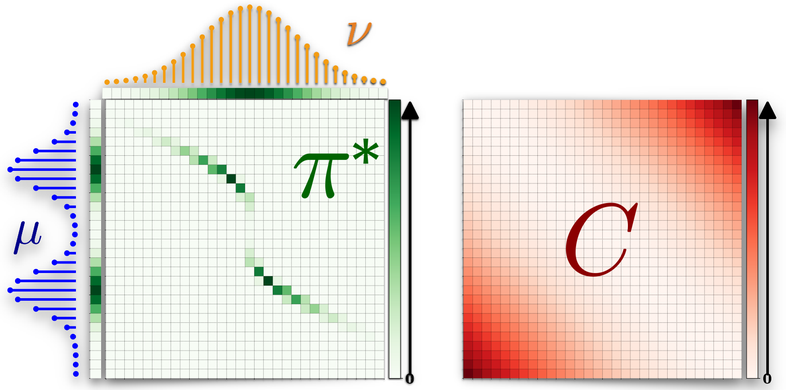
$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}, \qquad \nu = \sum_{j=1}^{m} b_j \delta_{y_j}, \qquad C_{ij} = c(x_i, y_j)$$



[1] Peyré and Cuturi, "Computational Optimal Transport", *Foundations and Trends in Machine Learning*, 2019.

Discrete Optimal Transport (OT)

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}, \qquad \nu = \sum_{j=1}^{m} b_j \delta_{y_j}, \qquad C_{ij} = c(x_i, y_j)$$

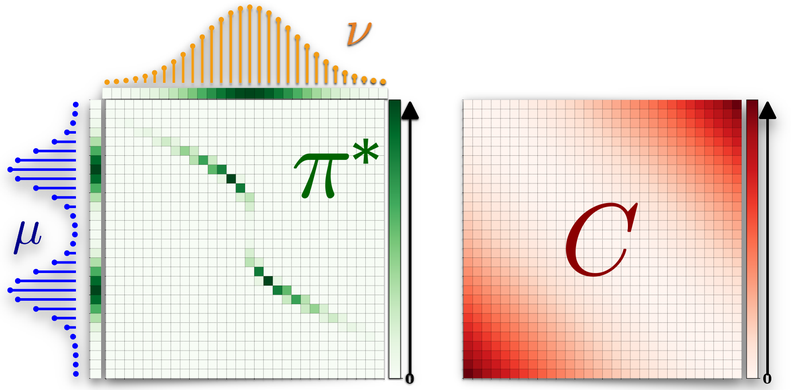


$$\mathrm{OT}_C(\mu, \nu) := \min_{\pi \in \mathbb{R}_+^{n \times m}} \left\{ \langle \pi, C \rangle = \sum_{i,j} \pi_{ij} C_{ij} \;\middle|\; \pi \mathbf{1}_m = a,\ \pi^\top \mathbf{1}_n = b \right\}$$

[1] Peyré and Cuturi, "Computational Optimal Transport", *Foundations and Trends in Machine Learning*, 2019.

Discrete Optimal Transport (OT)

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}, \qquad \nu = \sum_{j=1}^{m} b_j \delta_{y_j}, \qquad C_{ij} = c(x_i, y_j)$$



$$\mathrm{OT}_C(\mu,\nu) := \min_{\pi \in \mathbb{R}_+^{n\times m}} \left\{ \langle \pi, C \rangle = \sum_{i,j} \pi_{ij} C_{ij} \;\middle|\; \pi \mathbf{1}_m = a,\ \pi^\top \mathbf{1}_n = b \right\}$$

Property 2.1 (**Sparsity**). For the discrete Kantorovich problem $\mathrm{OT}_C(\mu,\nu)$, an optimal plan π^* exists with at most $n + m - 1$ nonzeros [Peyré and Cuturi, 2019, Prop. 3.4].

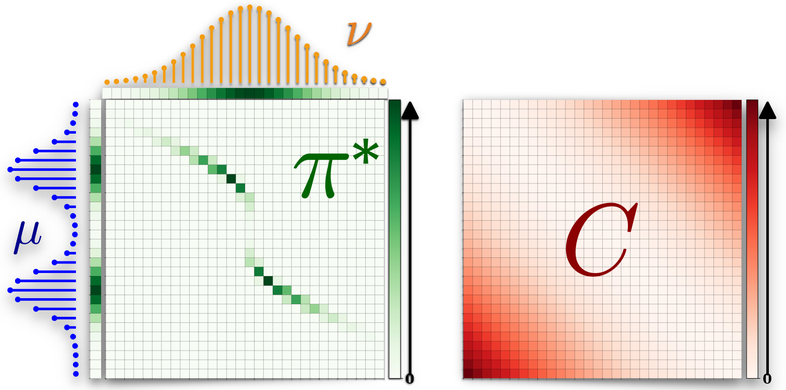
[1] Peyré and Cuturi, "Computational Optimal Transport", *Foundations and Trends in Machine Learning*, 2019.

Motivation: Why OT is a structured prediction problem

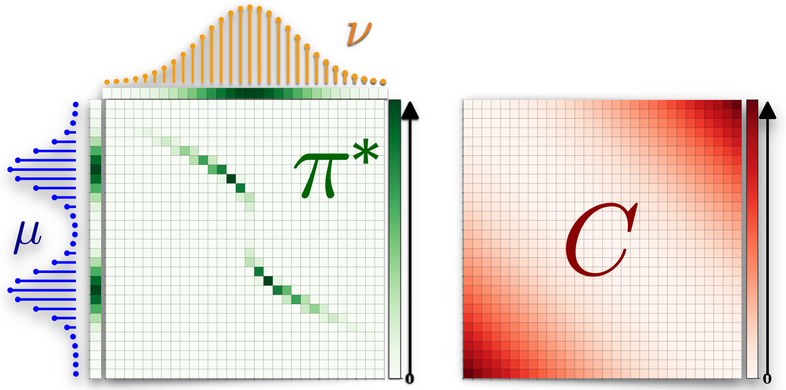
Classical OT solvers:

Solve for each $(\mu, \nu, C) \mapsto \pi^*$ from scratch.

Ignore shared structural knowledge from past problems.



Motivation: Why OT is a structured prediction problem



Classical OT solvers:

Solve for each $(\mu, \nu, C) \mapsto \pi^*$ from scratch.

Ignore shared structural knowledge from past problems.

OT as structured prediction:

Target output is a sparse coupling matrix under linear constraints.

Exact ground truth from classical solvers.

Sparse list instead of a dense $n \times m$ matrix:

$$\pi \equiv \{(i_k, j_k, w_k)\}_{k=1}^{K},\ K \leq n + m - 1$$

Research question:

Can autoregressive token prediction be adapted to learn to solve OT-type structured optimization problems when shown sufficient training data?

Token Regression: numbers as token sequences

Fixed vocabulary : $\mathscr{V} = \{0,\ldots,9,\ .,\ \ldots\}$

A target $y \in \mathbb{R}$ is written as a sequence of tokens: $y \mapsto (t_1, \ldots, t_K), \quad t_k \in \mathscr{V}$

[2] Song & Bahri, "Decoding-based regression", *TMLR*, 2025.

Token Regression: numbers as token sequences

Fixed vocabulary : $\mathscr{V} = \{0,\ldots,9,\ .,\ \ldots\}$

A target $y \in \mathbb{R}$ is written as a sequence of tokens: $y \mapsto (t_1, \ldots, t_K), \quad t_k \in \mathscr{V}$

Example: $y = 0.000127$

digit by digit: [0] [.] [0] [0] [0] [1] [2] [7] 8 tokens

exponent first: [E-4] [1] [2] [7] [0] 5 tokens

[2] Song & Bahri, "Decoding-based regression", *TMLR*, 2025.

Token Regression: numbers as token sequences

Fixed vocabulary : $\mathscr{V} = \{0,\ldots,9,\ .,\ \ldots\}$

A target $y \in \mathbb{R}$ is written as a sequence of tokens: $y \mapsto (t_1, \ldots, t_K), \quad t_k \in \mathscr{V}$

Example: $y = 0.000127$

digit by digit: [0] [.] [0] [0] [0] [1] [2] [7] 8 tokens

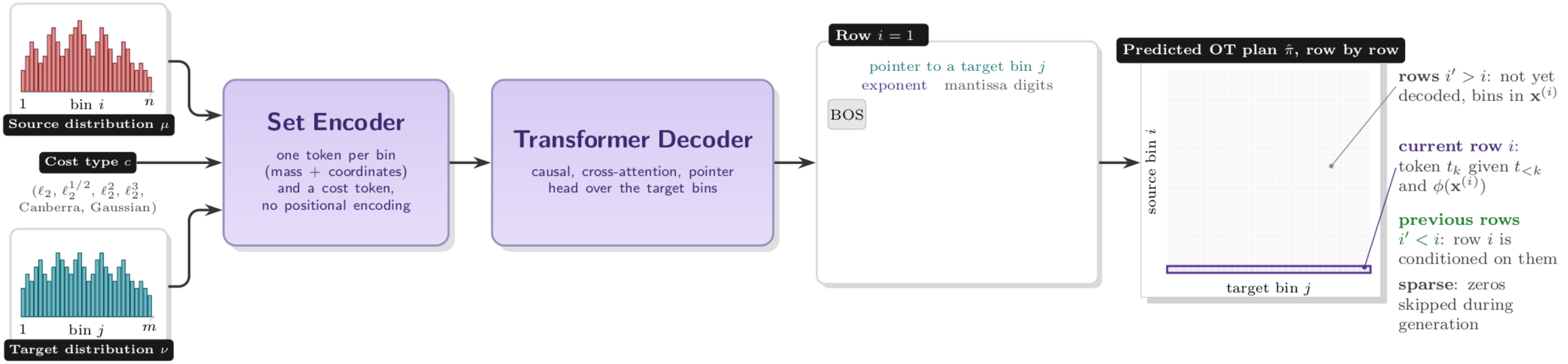
exponent first: [E-4] [1] [2] [7] [0] 5 tokens

A decoder generates one token at a time, conditioned on the input and previous tokens:

$$p_\theta(t_1, \ldots, t_K \mid \phi(\mathbf{x})) = \prod_{k=1}^{K} p_\theta\left(t_k \mid \phi(\mathbf{x}), t_{<k}\right) \qquad t_{<k} = (t_1, \ldots, t_{k-1})$$

[2] Song & Bahri, "Decoding-based regression", *TMLR*, 2025.

ToR-OT: **To**ken **R**egression **O**ptimal **T**ransport



Each nonzero: a pointer to target j, an exponent and mantissa digits (6 tokens total)

BOS $\left(\langle t_j \rangle\ \langle E_k \rangle\ d_1 d_2 d_3 d_4\right)^*$ EOS, $\qquad \hat{\pi}_{ij} = d_1 \,.\, d_2 d_3 d_4 \times 10^{-k}$

ToR-OT: Runtime Analysis

K_i counts the targets row i sends mass to, $\sum_{i=1}^{n} K_i \leq n + m - 1$

ToR-OT: Runtime Analysis

K_i counts the targets row i sends mass to, $\sum_{i=1}^{n} K_i \leq n + m - 1$

Decoder self-attention cost $\propto \sum_{i=1}^{n} K_i^2$

ToR-OT: Runtime Analysis

K_i counts the targets row i sends mass to, $\sum_{i=1}^{n} K_i \leq n + m - 1$

Decoder self-attention cost $\propto \sum_{i=1}^{n} K_i^2$

Typical case (approximately uniform sparsity)

$$K_i = \mathcal{O}(1) \implies \sum_i K_i^2 = \mathcal{O}(n)$$

ToR-OT: Runtime Analysis

K_i counts the targets row i sends mass to, $\sum_{i=1}^{n} K_i \leq n + m - 1$

Decoder self-attention cost $\propto \sum_{i=1}^{n} K_i^2$

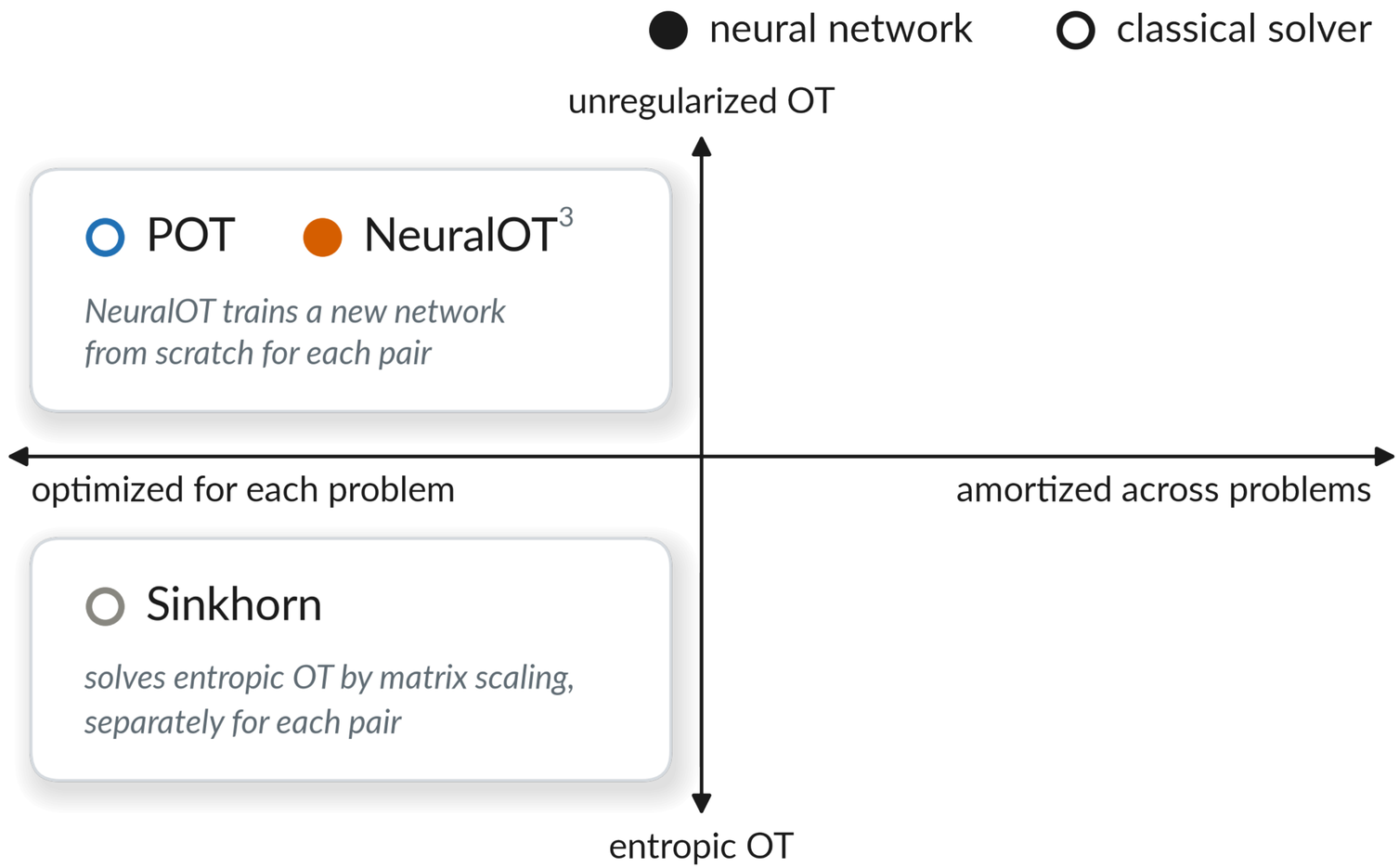
Typical case (approximately uniform sparsity)

$$K_i = \mathcal{O}(1) \implies \sum_i K_i^2 = \mathcal{O}(n)$$

Worst case (highly uneven sparsity)

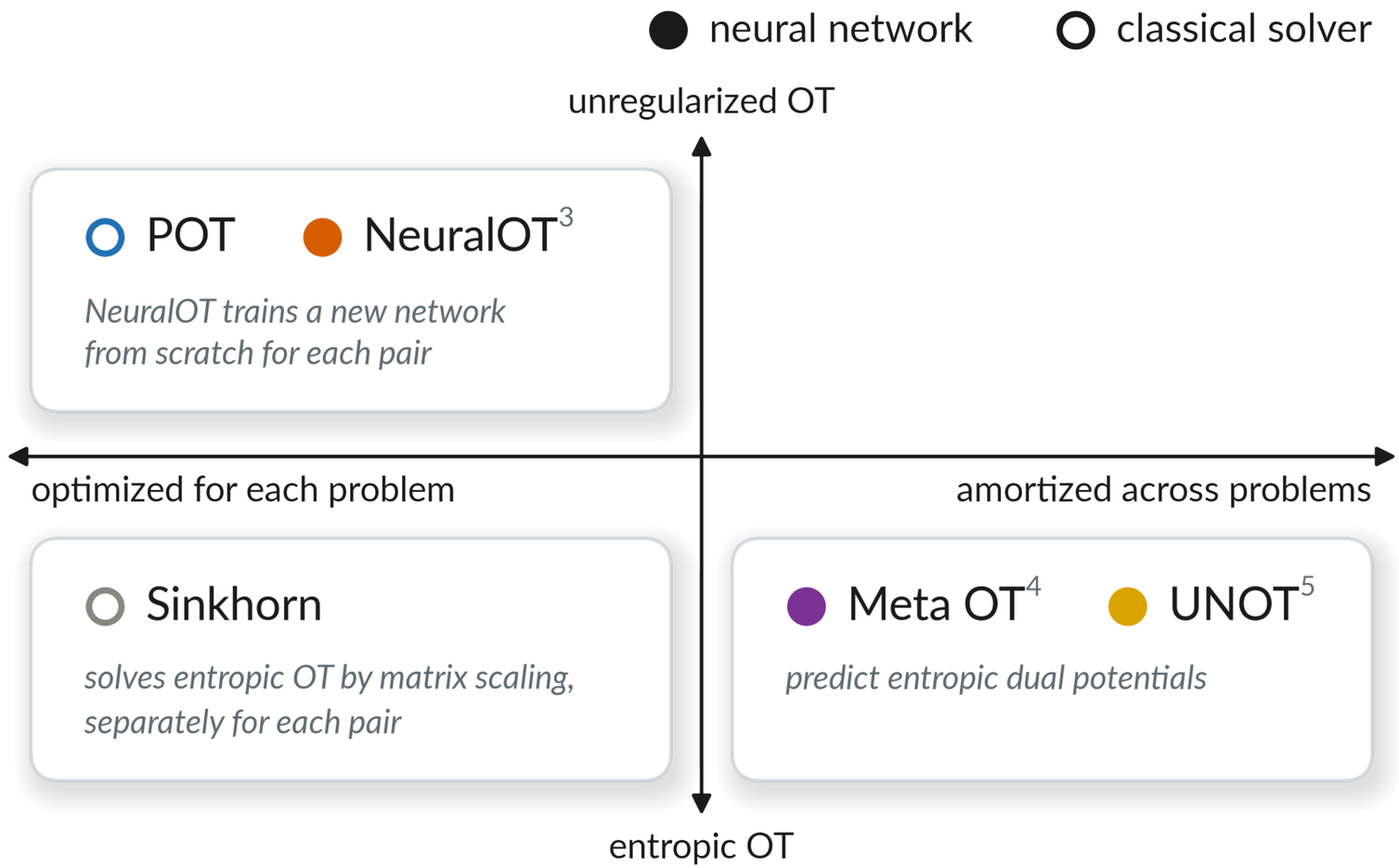
$$K_1 = \mathcal{O}(n) \implies \sum_i K_i^2 = \mathcal{O}(n^2)$$

ToR-OT: Positioning Among OT Solvers



[3] Korotin, Selikhanovych, and Burnaev, "Neural Optimal Transport", *ICLR*, 2023.

ToR-OT: Positioning Among OT Solvers

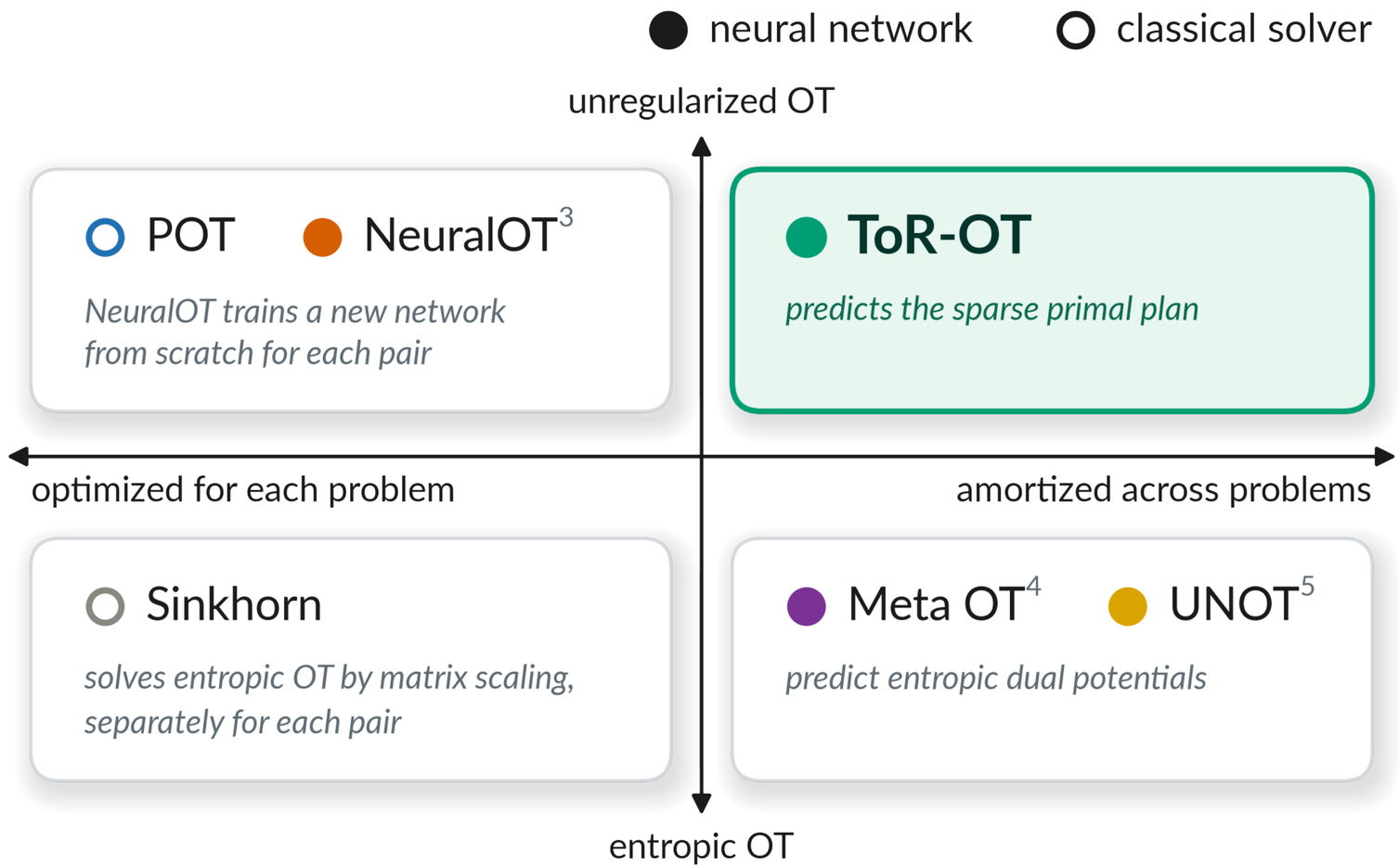


[3] Korotin, Selikhanovych, and Burnaev, "Neural Optimal Transport", *ICLR*, 2023.

[4] Amos, Cohen, Luise, and Redko, "Meta Optimal Transport", *ICML*, 2023.

[5] Geuter, Kornhardt, Tomasson, and Laschos, "Universal Neural Optimal Transport", *ICML*, 2025.

ToR-OT: Positioning Among OT Solvers



[3] Korotin, Selikhanovych, and Burnaev, "Neural Optimal Transport", *ICLR*, 2023.

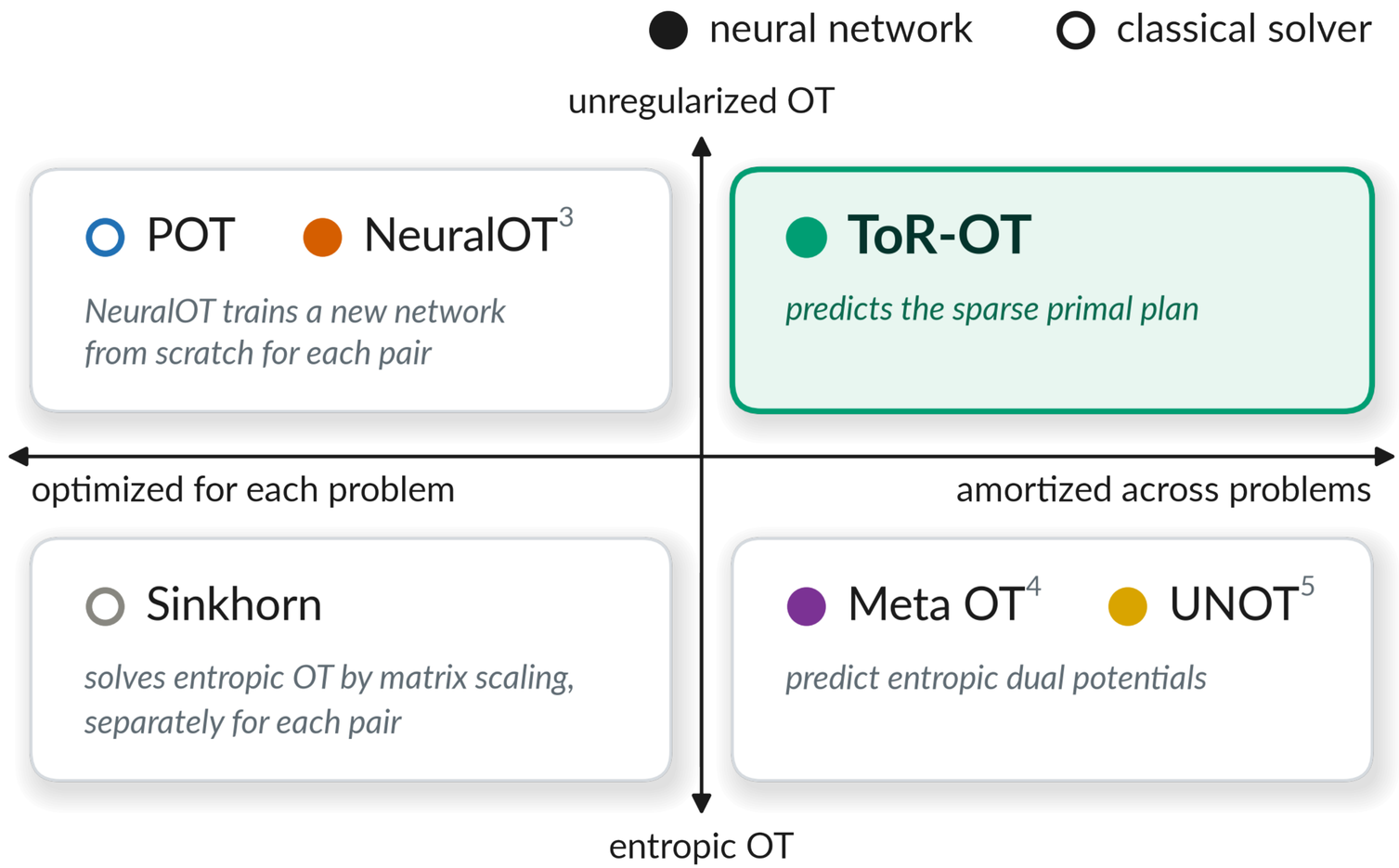
[4] Amos, Cohen, Luise, and Redko, "Meta Optimal Transport", *ICML*, 2023.

[5] Geuter, Kornhardt, Tomasson, and Laschos, "Universal Neural Optimal Transport", *ICML*, 2025.

ToR-OT: Positioning Among OT Solvers

When ToR-OT helps:

- Many similar problems recur.
- Warm-start solution.
- Unregularized sparse plan needed.



[3] Korotin, Selikhanovych, and Burnaev, "Neural Optimal Transport", *ICLR*, 2023.

[4] Amos, Cohen, Luise, and Redko, "Meta Optimal Transport", *ICML*, 2023.

[5] Geuter, Kornhardt, Tomasson, and Laschos, "Universal Neural Optimal Transport", *ICML*, 2025.

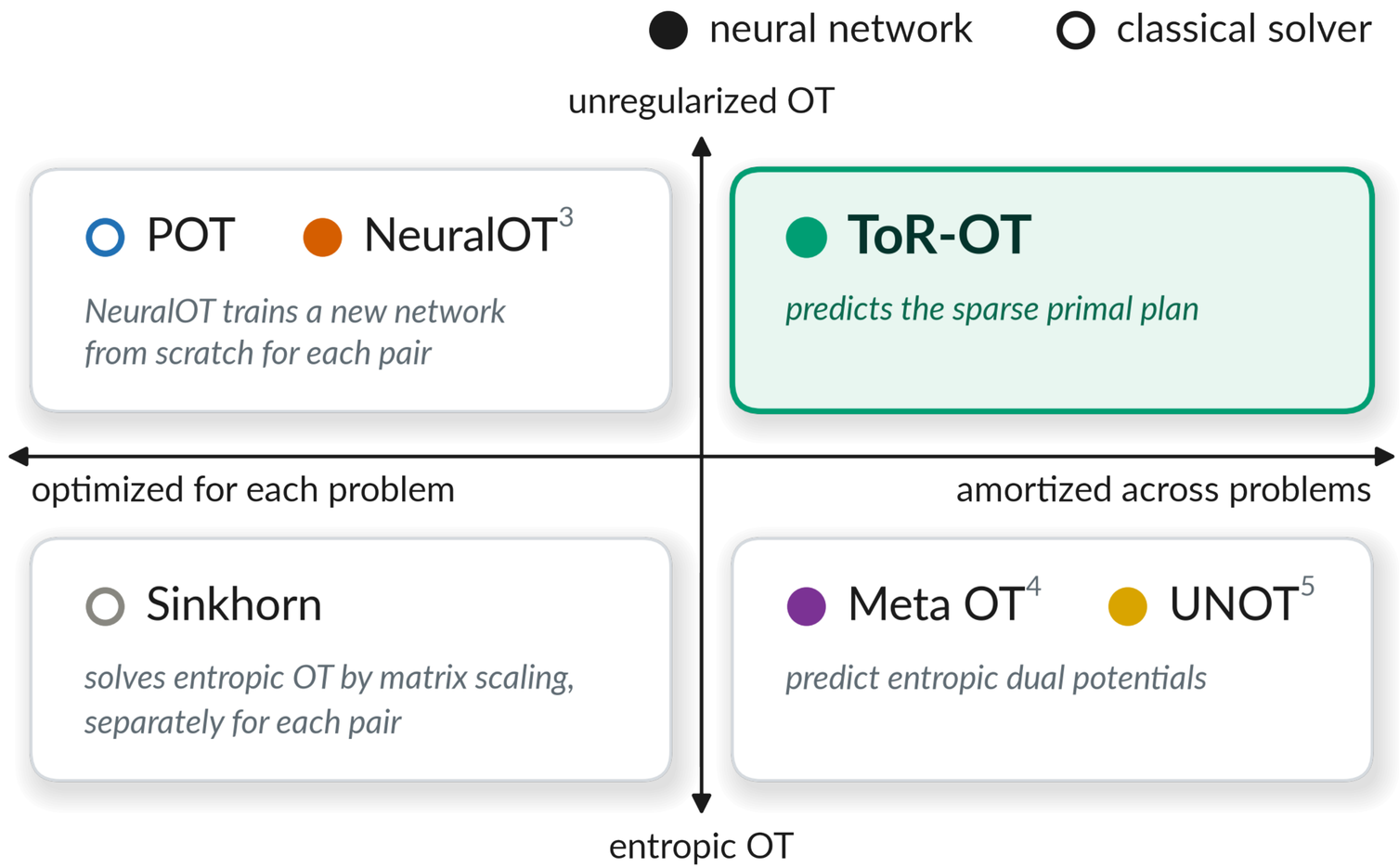
ToR-OT: Positioning Among OT Solvers

When ToR-OT helps:

- Many similar problems recur.
- Warm-start solution.
- Unregularized sparse plan needed.

When to use a classical solver:

- Certified optimality needed.
- Dense plans (e.g., entropic OT).



[3] Korotin, Selikhanovych, and Burnaev, "Neural Optimal Transport", *ICLR*, 2023.

[4] Amos, Cohen, Luise, and Redko, "Meta Optimal Transport", *ICML*, 2023.

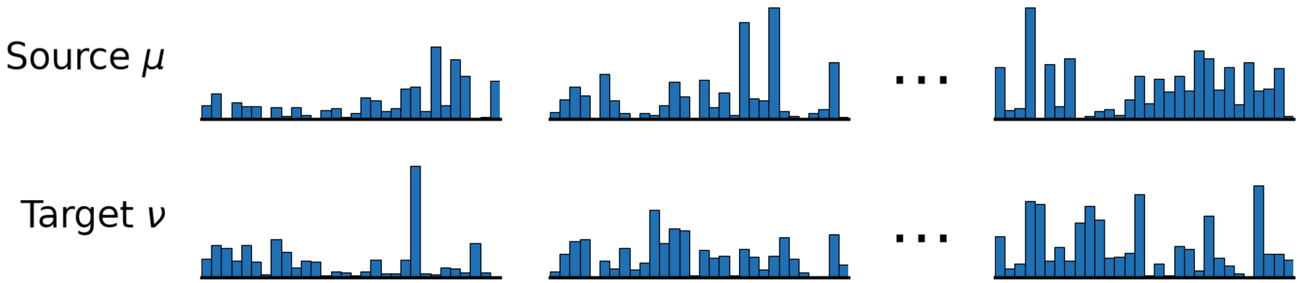
[5] Geuter, Kornhardt, Tomasson, and Laschos, "Universal Neural Optimal Transport", *ICML*, 2025.

Experimental Setup

Model:

Encoder-decoder Transformer.

Experimental Setup



Model:

Encoder-decoder Transformer.

Training data:

Synthetic random distributions.

Experimental Setup

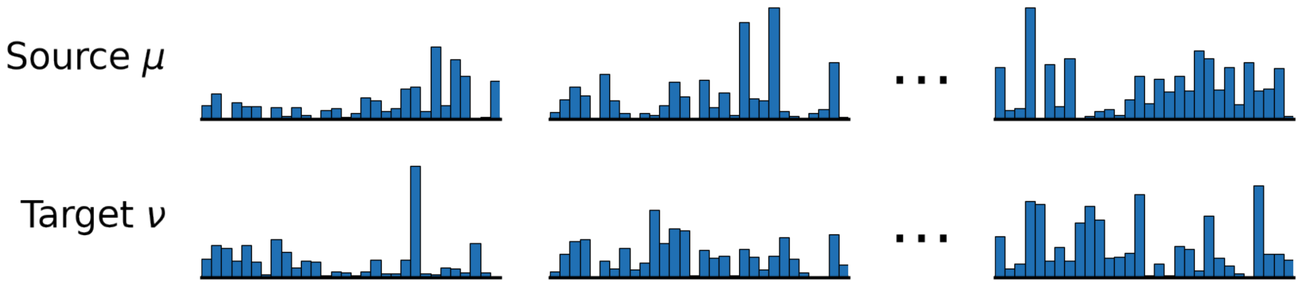
Model:

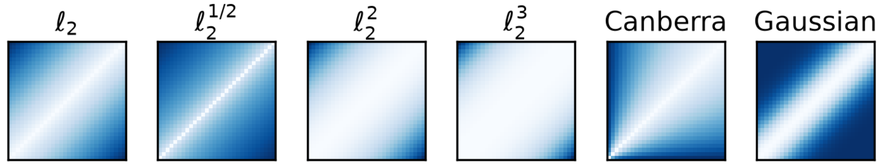
Encoder-decoder Transformer.

Training data:

Synthetic random distributions.

6 cost types.





Experimental Setup

Model:

Encoder-decoder Transformer.

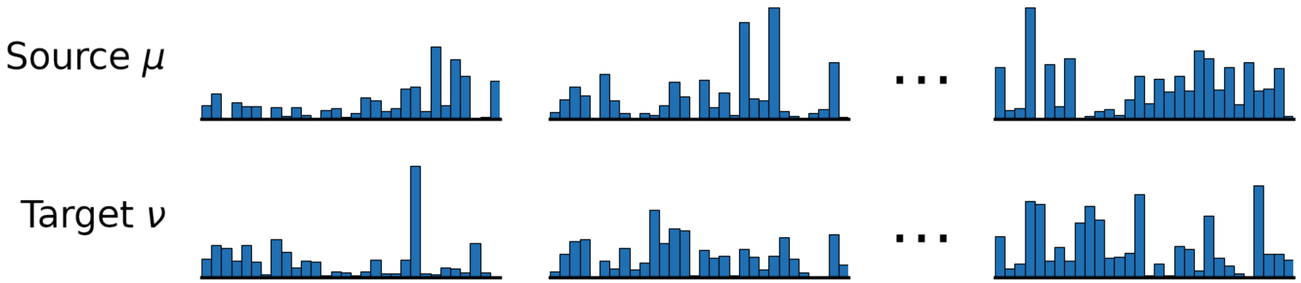
Training data:

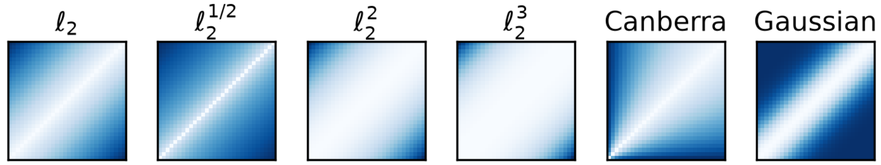
Synthetic random distributions.

6 cost types.

Baselines:

Neural OT, Meta OT, UNOT.





Experimental Setup

Model:

Encoder-decoder Transformer.

Training data:

Synthetic random distributions.

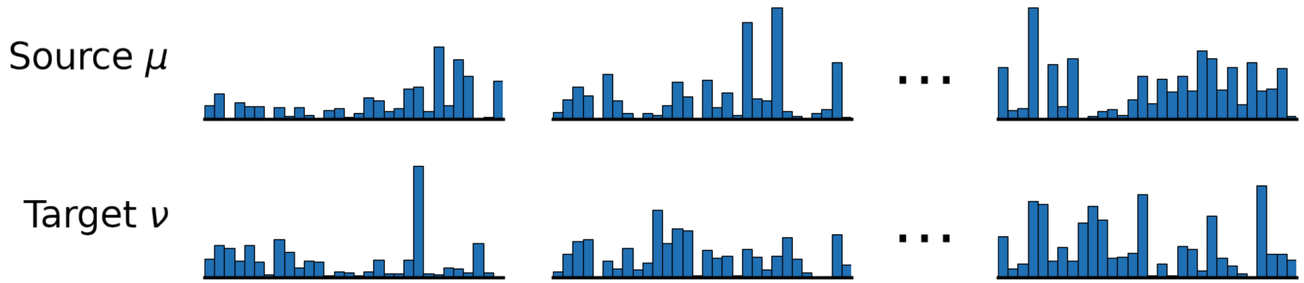
6 cost types.

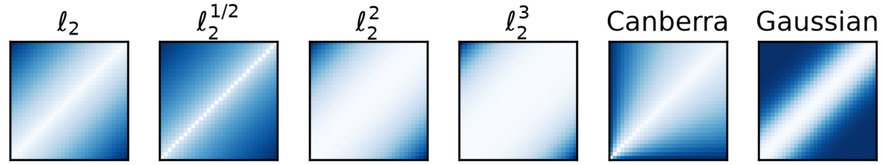
Baselines:

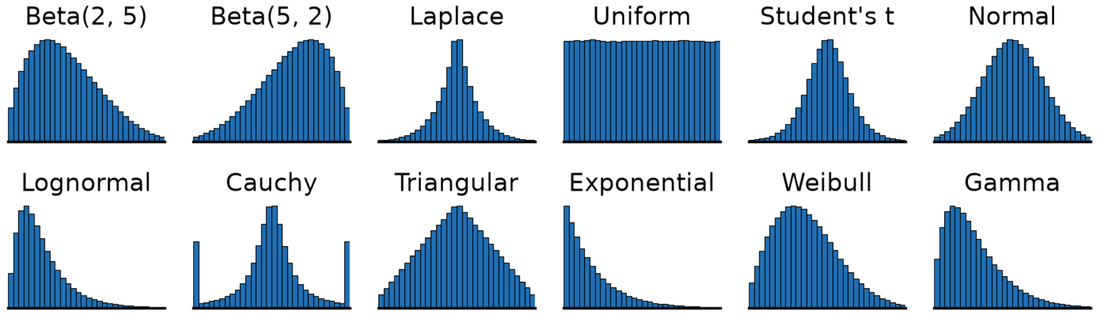
Neural OT, Meta OT, UNOT.

Evaluation:

Test and OOD sets.

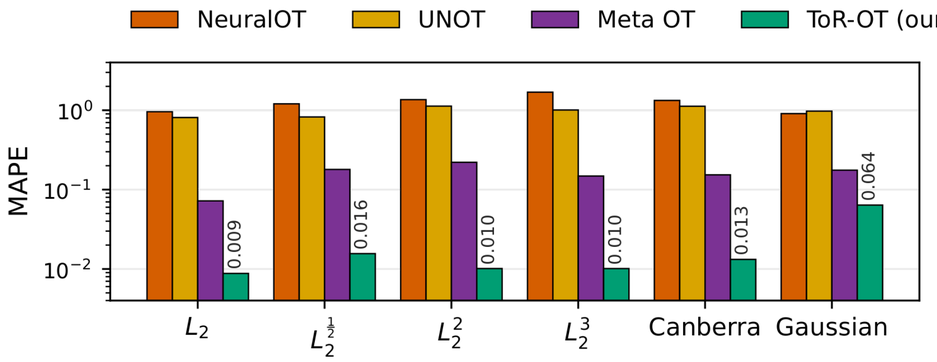


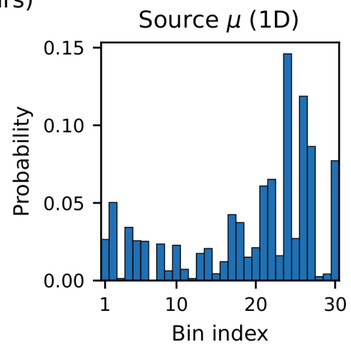


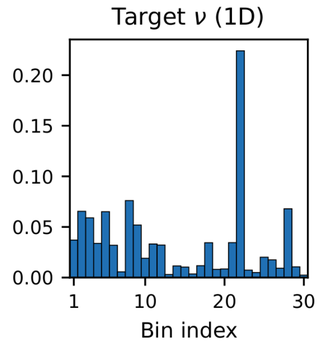


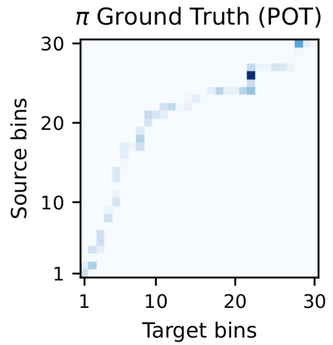
Empirical Results on 1D distributions

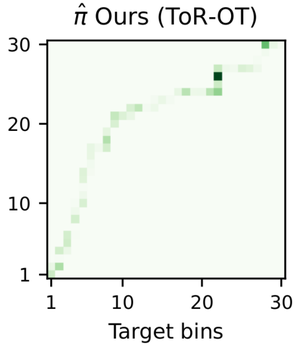
Test set



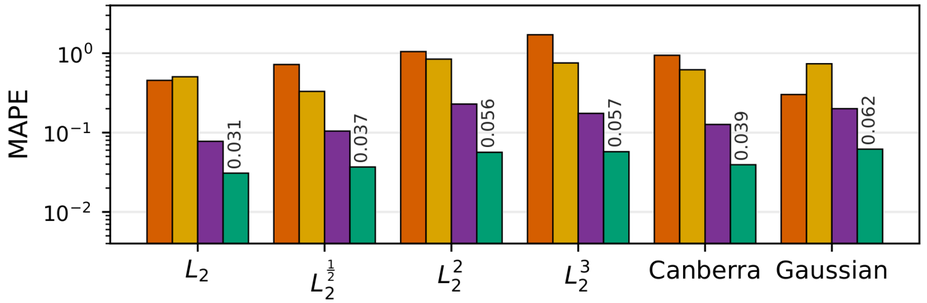


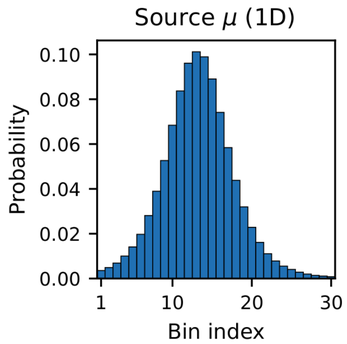


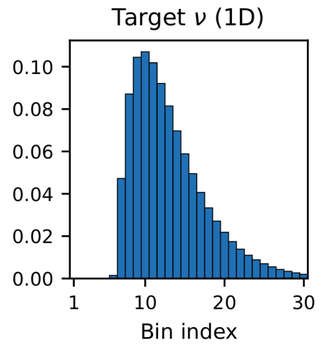


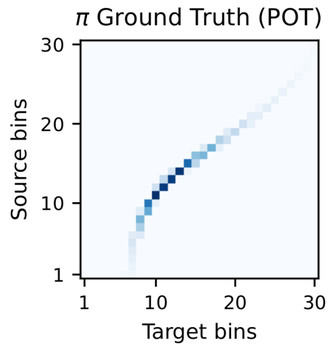


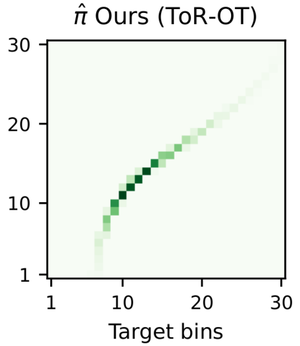
OOD set











Lowest mean error on all 6 costs: 64–95% (test set) below Meta OT, 60–75% (OOD set).

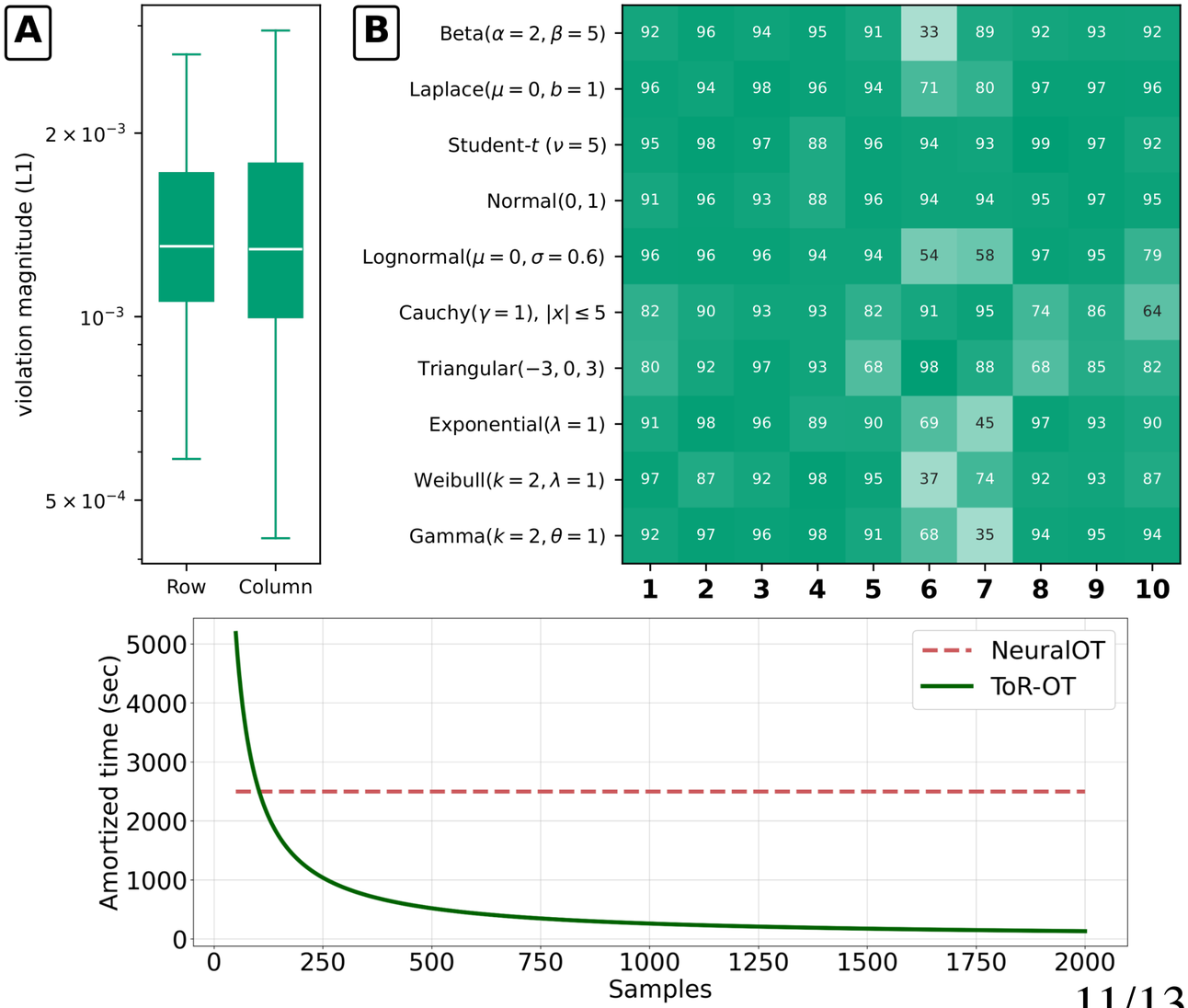
Predicted plan matches the exact plan's sparse structure.

Empirical Results on 1D distributions

A. Feasibility:

Median violation ≈ 0.1% of total mass

Column violations vary more than row violations.



Empirical Results on 1D distributions

A. Feasibility:

Median violation ≈ 0.1% of total mass

Column violations vary more than row violations.

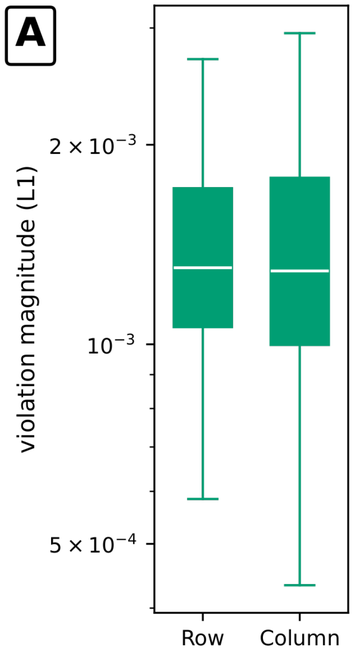
B. OOD set, per pair (ℓ_2^2 cost) vs Neural OT:

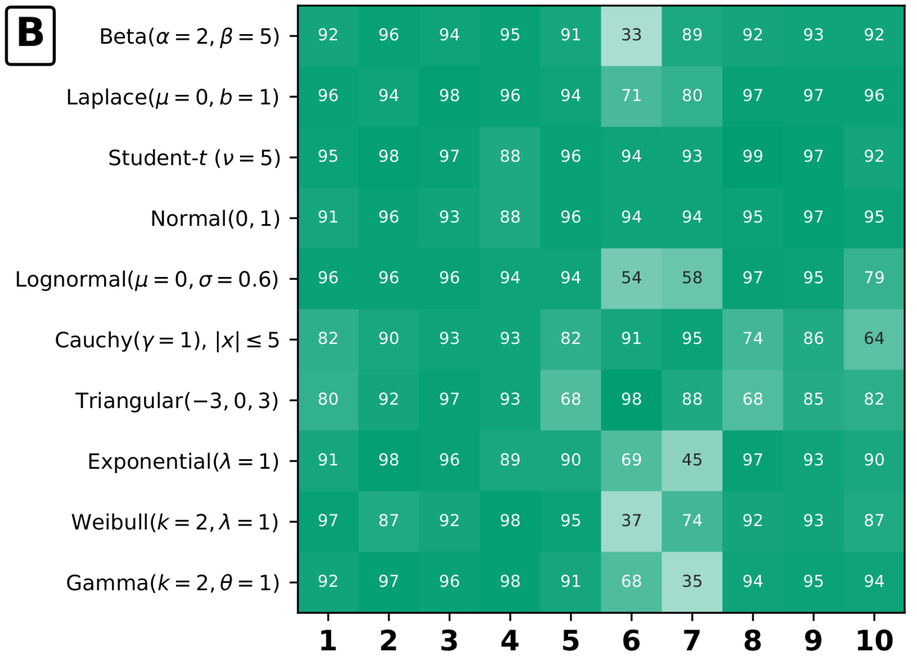
Median improvement 91% (mean MAPE 75% lower).

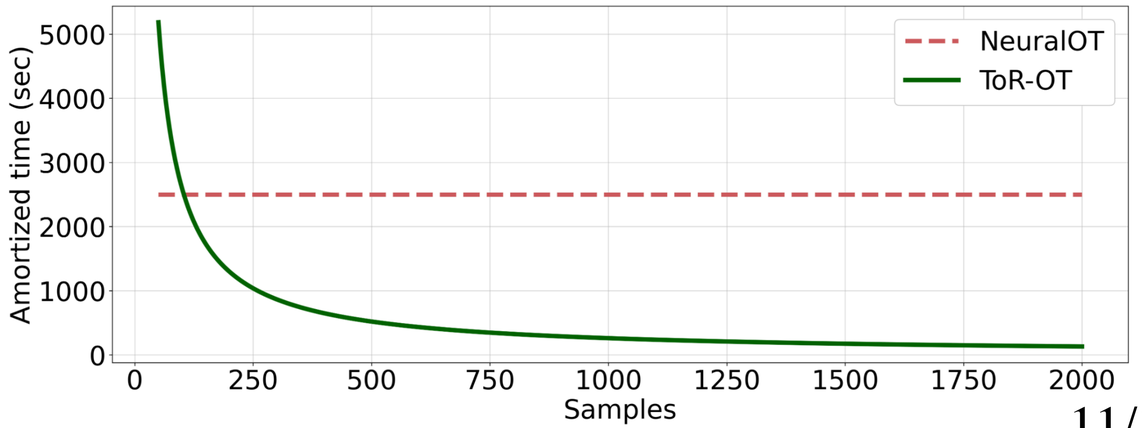
Runtime vs Neural OT:

1.09 s per plan on one A100 GPU

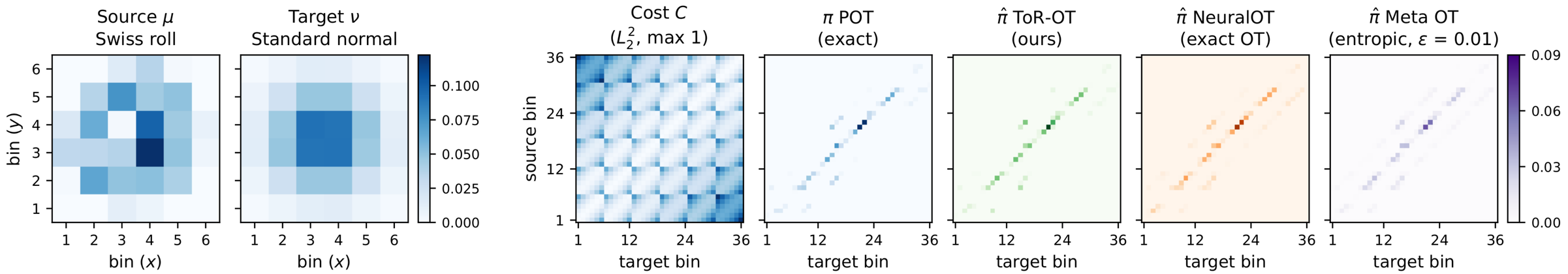
Training cost (24 h) pays off in under 100 problems.





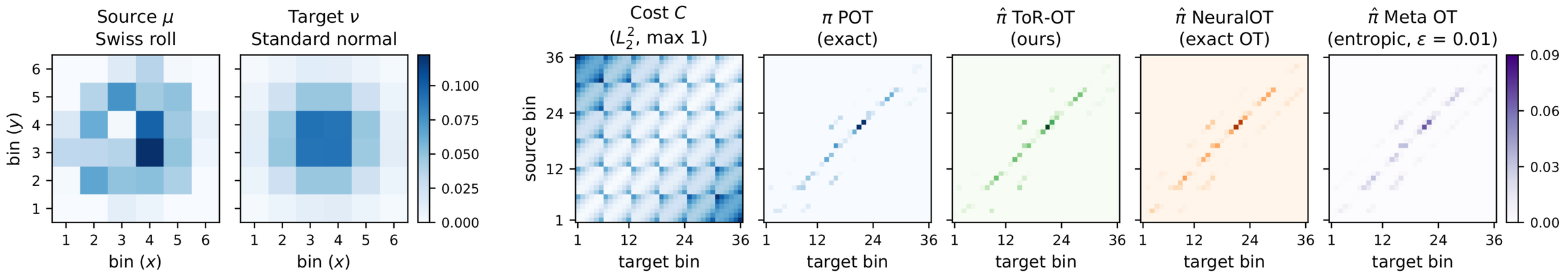


Empirical Results on 2D distributions



Same model architecture and training routine.

Empirical Results on 2D distributions



Same model architecture and training routine.

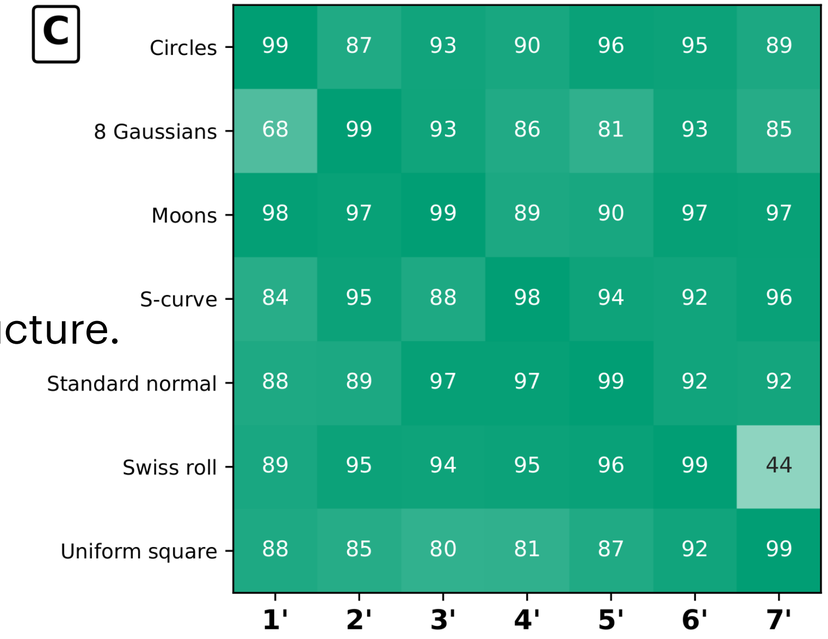
Trained on 4M pairs (k×k grids, k = 3–10).

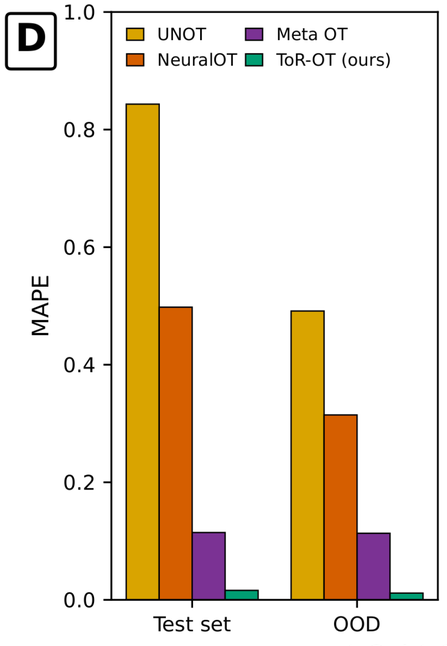
Tested on 6×6 (n = m = 36).

Top row. ToR-OT follows the exact plan's sparse structure.

C. Beats Meta OT on all 49 OOD pairs.

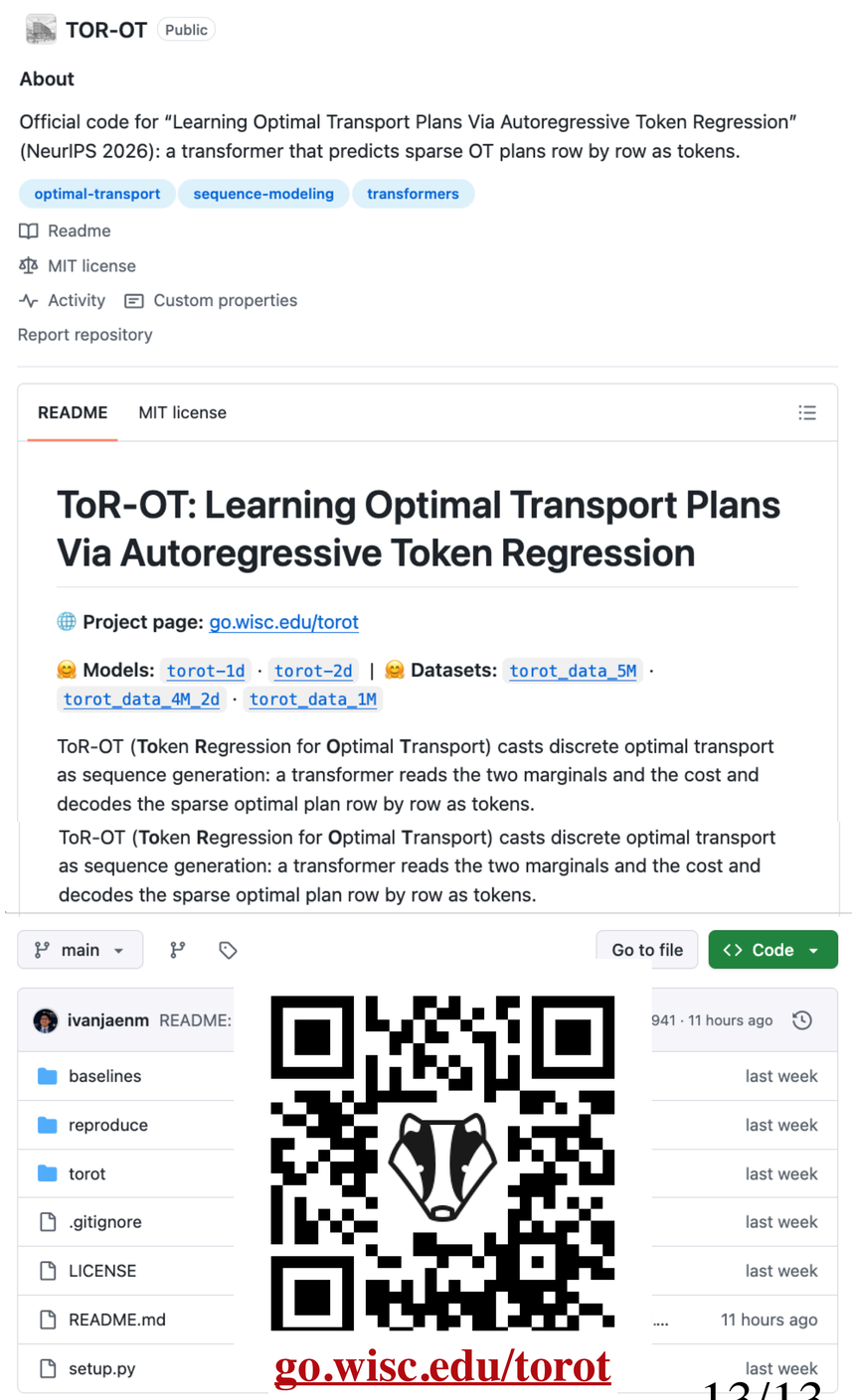
D. Lowest mean MAPE on the test and OOD sets.





Conclusion

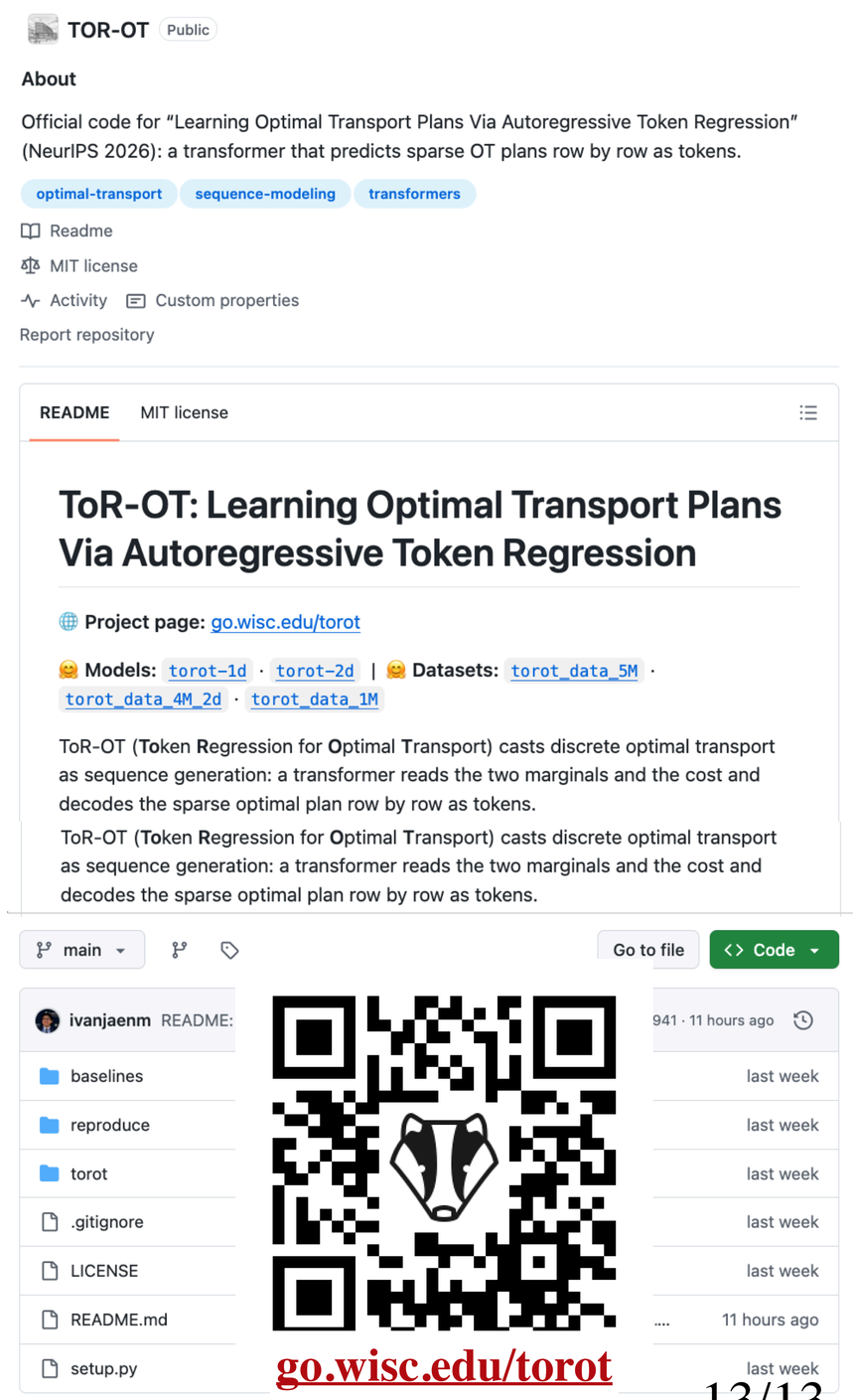
ToR-OT treats OT plan prediction as sequence generation rather than numerical optimization.



Conclusion

ToR-OT treats OT plan prediction as sequence generation rather than numerical optimization.

Unregularized primal OT plans can be learned by sequence models by exploiting sparsity.

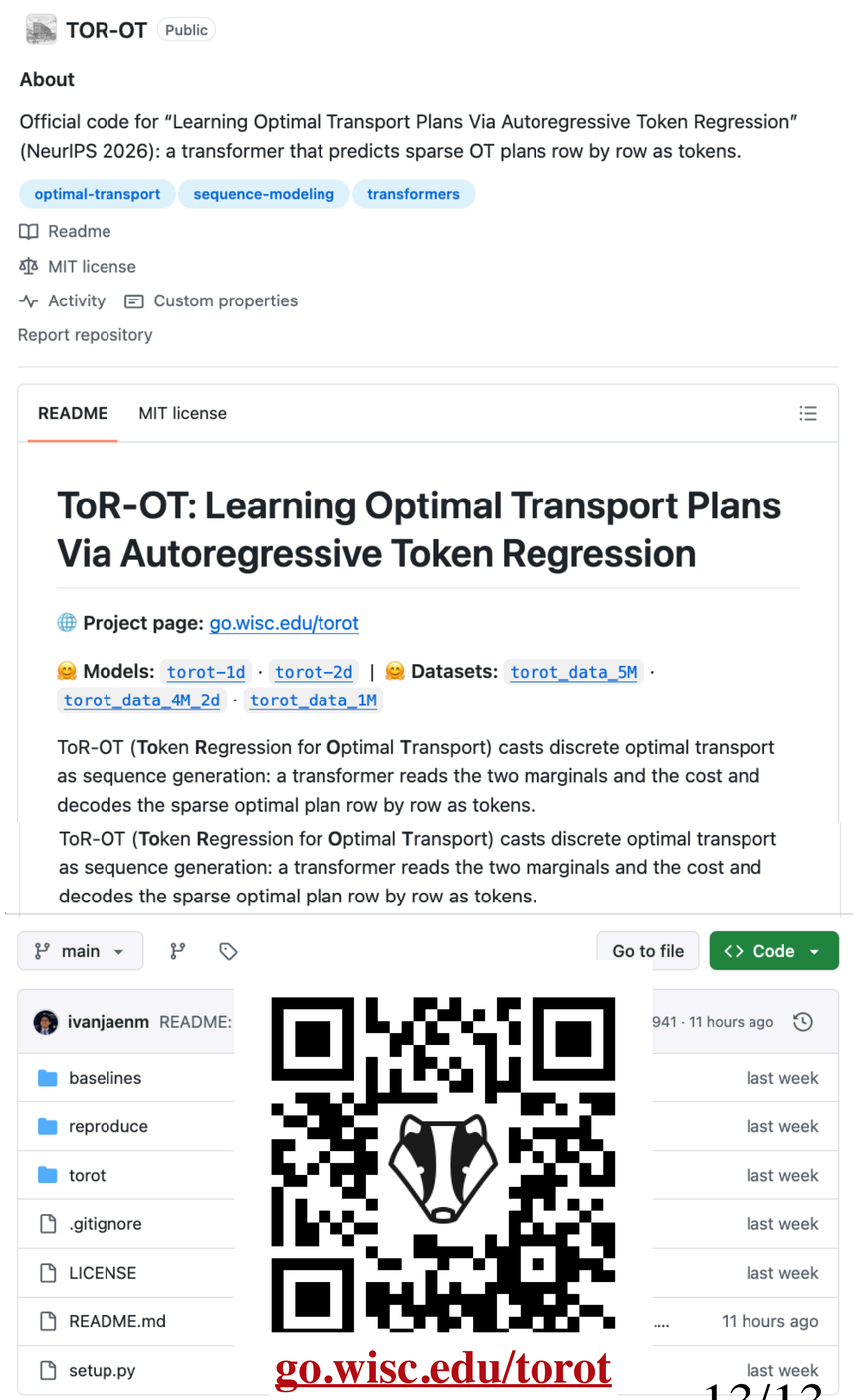


Conclusion

ToR-OT treats OT plan prediction as sequence generation rather than numerical optimization.

Unregularized primal OT plans can be learned by sequence models by exploiting sparsity.

A single pre-trained model provides accurate plans on unseen distributions and multiple costs.



Conclusion

ToR-OT treats OT plan prediction as sequence generation rather than numerical optimization.

Unregularized primal OT plans can be learned by sequence models by exploiting sparsity.

A single pre-trained model provides accurate plans on unseen distributions and multiple costs.

A step toward the study of Transformers for OT, not yet a replacement for classical solvers.

