Placeto: Efficient Progressive Device Placement Optimization

Ravichandra Addanki, Shaileshh Bojja Venkatakrishnan, Shreyan Gupta, Hongzi Mao, Mohammad Alizadeh

University of Minnesota
Driven to Discover℠

Recall--- What is Device Placement

- G(V,E): the computational graph of a neural network
- D: a set of devices (e.g., CPUs, GPUs)
- π: V -> D
- p(G,π): the duration of G's execution when its ops are placed according to π
- Goal: find a placement π that minimizes p(G,π)

Recall --- Why need Device Placement

- Trend toward many-device training, bigger models, larger batch sizes

- Growth in size and computational requirements of training and inference

Recall --- Current Approach

- Human Expert

  (1) Require deep understanding of devices (e.g., bandwidth & latency behavior);

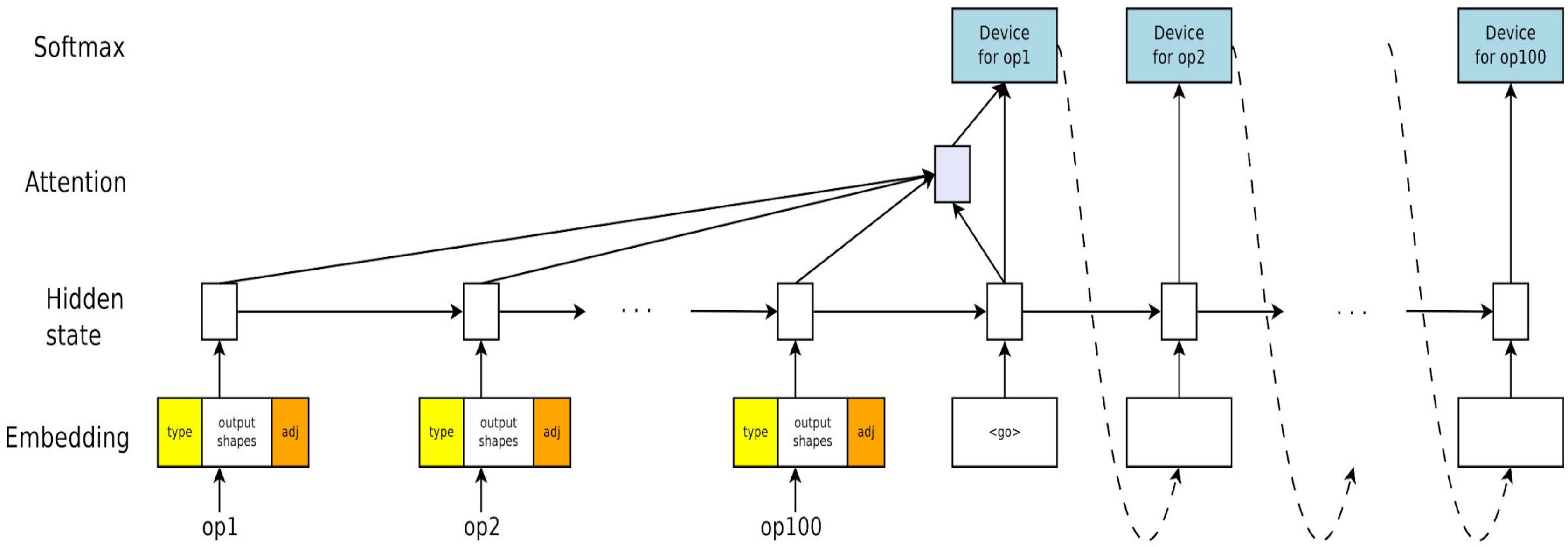
  (2) Not flexible enough & not generalize well.

- Automated Approach (RNN-based Approach)

  (1) Require significant amount of training/training time is long (e.g., 12-27 hours);

  (2) Do not learn generalizable device placement policies.

Recall --- RNN-based Approach



Can it be better?

- Is it able to transfer a learned placement policy to unseen computational graphs without extensive re-training?
- Is it possible to improve training efficiency and generalizability?

Placeto --- Key Ideas

- Model the device placement task as finding a sequence of iterative placement improvements

- Use Graph Embeddings to encode the computational graph structure

Design --- MDP Formulation

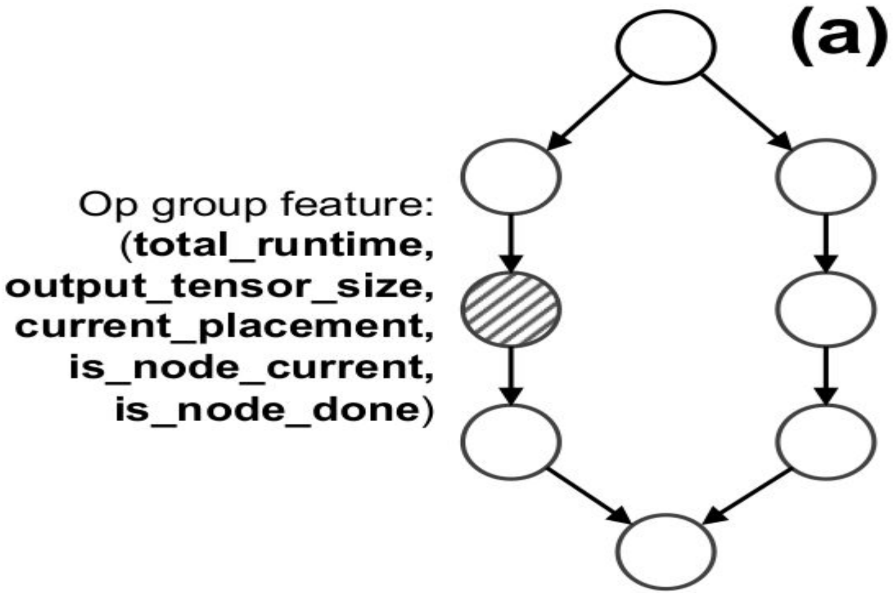
- Initial state s_0, consists of G with an arbitrary device placement for each op group
- Action in step t outputs a new placement for the t-th node in G based on s_{t-1}
- Episode ends in |V| steps
- Two approaches for assigning rewards:

(1) Assign 0 reward at each intermediate RL step & the negative run time of the final replacement as final reward

(2) Assign intermediate rewards $r_t = p(s_{t+1}) - p(s_t)$

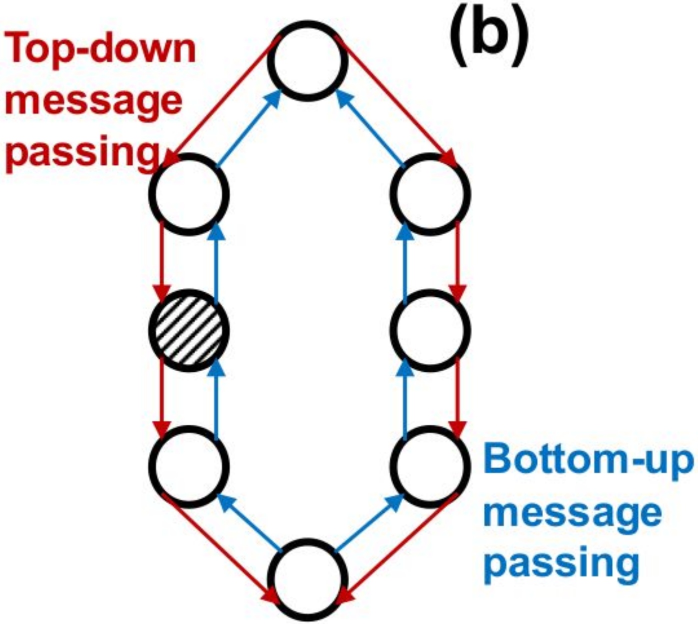
Design --- Graph Embedding (1/3)

- Computing per-group attributes



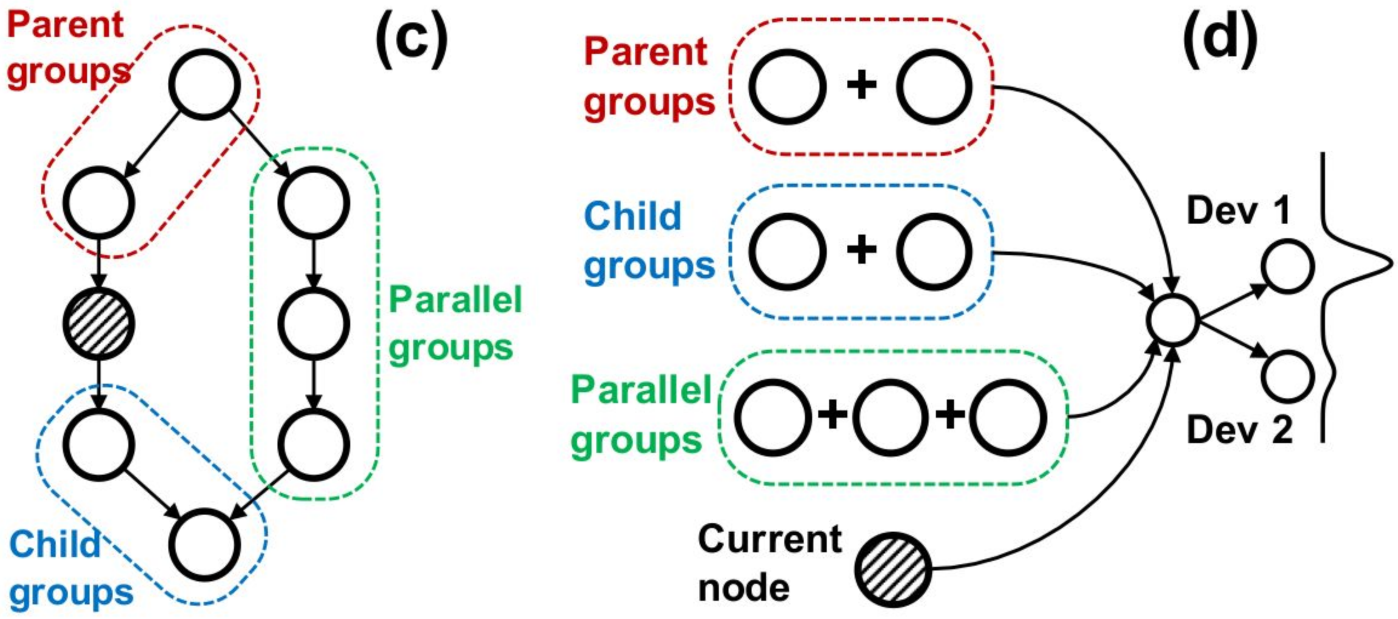
Design --- Graph Embedding (2/3)

- Local neighborhood summarization



Design --- Graph Embedding (3/3)

- Pooling summaries



Experiments

- How good are Placeto's placements in terms of execution time?

- How well does Placeto generalize to unseen graph?

Experiments

- Benchmark computational graphs:

  (1) Inception-V3

  (2) NASNet

  (3) NMT

- Baseline:

  (1) Human-expert placement

  (2) RNN-based approach

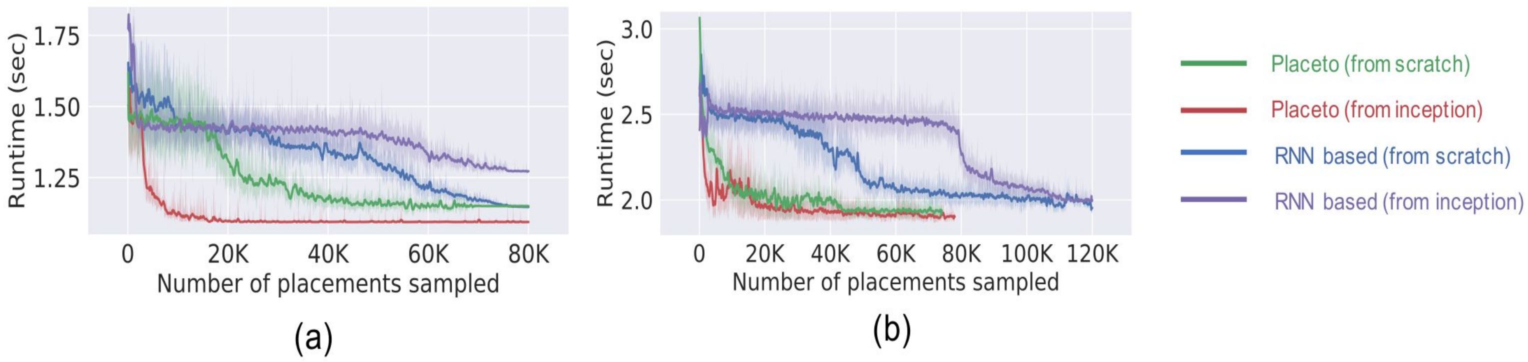
Experiments

- Performance

	Placement run time (sec)				Training time (# placements sampled)			
Model	Expert	RNN-based [10]	Placeto (scratch)	Placeto (transfer)	RNN-based [10]	Placeto (scratch)	Placeto (transfer)	Speedup factor
Inception-V3	1.27	1.21	1.20	—	11.2 K	3.7 K	—	—
NMT	2.00	1.52	1.52	1.57	84 K	20 K	3.9 K	**21**×
NASNet	0.86	0.84	0.83	0.84	76 K	28.8 K	12.2 K	**6**×

Experiments

- Generalizability



Future Work

- Using a mix of models with diverse graph structures during training, Placeto may exhibit better generalizability.
- Larger graphs, larger batch sizes, and more heterogeneous will be more challenging and can potentially lead to larger gains.
- Extend Placeto to jointly learn ops grouping and placement.