The University of Texas at Austin
Cockrell School of Engineering

# Goal-Conditioned Supervised Learning for Multi-Objective Recommendation

Shijun Li, Hilaf Hasson, Jin Hu, Joydeep Ghosh

**Presented by Shijun Li**
Department of Electrical and Computer Engineering, The University of Texas at Austin
11/13/2026

- Challenges of Multi-Objective Learning in Industrial Recommender Systems

  - Previous methods focuses on balancing objectives through complex architectural or optimization designs.

  - Large amount of existing sequential data on multiple feedback types naturally provides valuable learning signals.

- Goal-Conditioned Supervised Learning (GCSL)

Learn a next action suggestion model, which can suggest the next action that is mostly likely to achieve a given goal.

- Strengths
  - With above policy, we can expect a generated sequence of actions by simply giving an initial state and a desired goal.
  - No special structure and optimization designs with high complexity are further required.
  - Faster convergence.

- What do goals mean?

- Goals are what we achieved or what we want to achieve from current timestep to the final step during a sequence.

- One straightforward definition:
  $g = R_t = \sum_{t'=t}^{T}[\gamma^{t'} * r_{t'}]$ , which is exactly the cumulative reward.



The company's projected revenue for the next month is 1 million dollars

The goal is accordingly defined:

$$g = \sum_{t=1}^{30}[revenue_t] = 10^6$$

- Adaption for multi-objective learning

  - Design a vectorized goal, where each dimension denotes a certain objective:

$$g_t = [\sum_{t'=t}^{end} click_{t'}, \sum_{t'=t}^{end} buy_{t'}, \dots]$$

  - Then the problem should be formalized as a Multi-Objective Markov Decision Process problem, where each dimension of the goal represents the cumulative reward associated with a specific objective

- Offline training

Considering a typical offline data sequence:

$$\{(s_1, a_1, r_1), (s_2, a_2, r_2), \dots, (s_T, a_T, r_T)\}$$

- Calculate the goals at each timestep:

$$g_t = \sum_{t'=t}^{T} [\gamma^{t'} * r_{t'}]$$

- Reprocess the offline data:

$$D = \{(s_1, a_1, g_1), (s_2, a_2, g_2), \dots, (s_T, a_T, g_T)\}$$

- Train to maximize likelihood with supervised learning:

$\pi = argmax_{\pi_\theta} \mathrm{E}_D[log\pi_\theta(a|s, g)]$, where $s = \{s_1, \dots, s_t\}$

- Online inference

- Original use: First specify a desired goal, then generate the sequence based on the goal and initial state

- Our approach: Automatically choose an "achievable and good" goal for each user, so that the suggested next action will be conditioned on individual inference instances.

- We provided the first theoretical analysis of the achievable goal distribution in inference for multi-objective setting in GCSL by stating and proving a relevant theorem

**Theorem 1.** *Assume that the environment is modeled as an MOMDP. Consider a trajectory $\tau$ that is generated by the policy $\pi(a|\boldsymbol{s}, \boldsymbol{g})$ given the initial state $\boldsymbol{s}_1$ and goal $\boldsymbol{g}_1$, the distribution of goals $\boldsymbol{g}^a$ (i.e., cumulative rewards) that the agent actually achieves throughout the trajectory is determined by $(\boldsymbol{s}_1, \boldsymbol{g}_1, \pi)$.*

- Based on it, we proposed a goal-choosing algorithm that models the distribution of achievable goals over interaction sequences and choose desirable “high” goals for inference

- We propose to use conditional variational auto-encoder (CVAE) to model and learn the distribution of the achievable goal $P(g^a|s', g', \pi)$.

- On inference, we can derive the estimation of the achievable goals based on this distribution for different input goal.

**Algorithm 2:** Inference of MOGCSL

**Input:** state $\boldsymbol{s}'$, sample size $K$, policy model $\pi$, utility principle $U$, prior $q(\boldsymbol{g}'|\boldsymbol{s}')$, distribution of achievable goals $P(\boldsymbol{g}^a|\boldsymbol{s}', \boldsymbol{g}', \pi)$
**Intialization:** set of potential input goals $\boldsymbol{G}' = \emptyset$, set of expected achievable goals $\boldsymbol{G}^a = \emptyset$
**for** $k = 1, \ldots, K$ **do**
    Sample a $\boldsymbol{g}'_k$ from $q(\boldsymbol{g}'|\boldsymbol{s}')$
    Compute the expectation of the achievable goal through sampling: $\tilde{\boldsymbol{g}}^a_k = \mathbb{E}_{\boldsymbol{g}^a_k \sim P(\cdot|\boldsymbol{s}', \boldsymbol{g}'_k, \pi)} \boldsymbol{g}^a_k$
    $\boldsymbol{G}' = \boldsymbol{G}' \cup \boldsymbol{g}'_k$
    $\boldsymbol{G}^a = \boldsymbol{G}^a \cup \tilde{\boldsymbol{g}}^a_k$
Choose the best $\tilde{\boldsymbol{g}}^a_b$ from $\boldsymbol{G}^a$ according to $U(\tilde{\boldsymbol{g}}^a)$
Choose corresponding $\boldsymbol{g}'_b$ from $\boldsymbol{G}'$
**Return:** $\pi(\cdot|\boldsymbol{s}', \boldsymbol{g}'_b)$

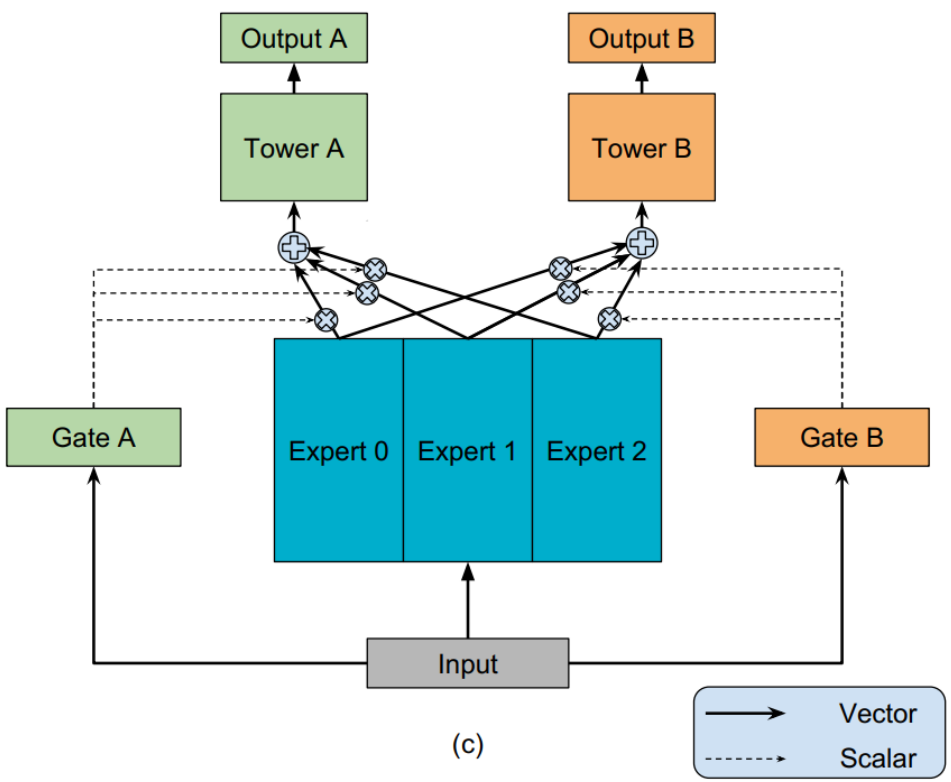
- How MOGCSL addressed complexity issues?

## Existing challenges

- Necessity of designing a distinct tower and output layer for each task escalates the complexity of the model when handling multiple tasks

- It's not clear how to constrain the learning process between multiple objectives and how to decide the weights of the multi-task loss function

- Previous method (MMoE)




- Each tower outputs prediction for one task, and then calculates a loss function on the prediction and the labels:

$$L_i = E_D[loss(y_i, \hat{y}_i)]$$

- Then, it's optimized by a holistic loss function by weighted summing the losses on each task:

$$L = \sum_i w_i * L_i$$

*: Ma, Jiaqi, et al. "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts." SIGKDD 2018.

## Advantages of MOGCSL

- Alleviate the complexity issue in a concise way. Directly output the final recommendation without the multi-tower design.

- The optimization quandary like tuning the loss weight combination is alleviated. The objective becomes learning the appropriate action to take in order to achieve certain performance on all objectives.

- Experiments

- Evaluate performance on public sequential recommendation datasets.

  - Metrics: Hit Ratio@k, NDCG@k
  - Baselines:
    - Existing multi-objective learning methods: Share-Fix, Share-DWA, Share-PE, MMOE-Fix, MMOE-DWA, MMOE-Nash.
    - Goal-conditional RL with weighted sum goals: PRL
      (e.g., $g = \sum_t [click * 0.2 + buy]$, choose goals as 2 times the mean goal)
    - Multi-objective recommendation methods: RMTL, PMORS

# • Experimental Results

- Goal-conditional Supervised Learning and disentangle task representations with vectorized goals can significantly improve the performance

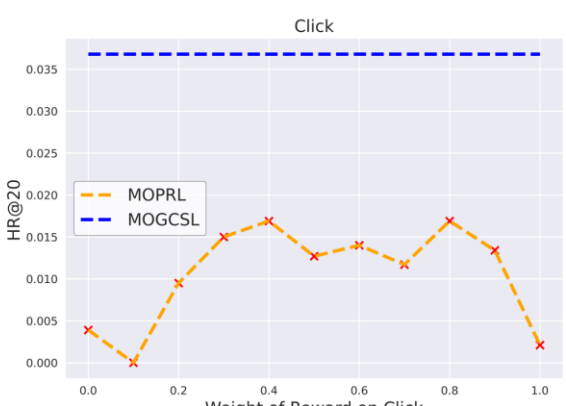
| [RetailRocket] | Purchase (%) | | | | Click (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | HR@10 | NG@10 | HR@20 | NG@20 | HR@10 | NG@10 | HR@20 | NG@20 |
| Share-Fix | 48.57±0.17 | 45.79±0.09 | 49.47±0.10 | 46.01±0.08 | 35.51±0.24 | 25.85±0.16 | 40.15±0.20 | 27.03±0.14 |
| Share-DWA | 48.11±0.04 | 45.83±0.08 | 48.64±0.08 | 45.96±0.06 | 34.20±0.27 | 25.53±0.12 | 38.43±0.31 | 26.60±0.13 |
| Share-PE | 48.67±0.12 | 45.94±0.03 | 49.42±0.04 | 46.13±0.02 | 35.69±0.08 | **26.20±0.08** | 40.28±0.11 | **27.37±0.07** |
| Share-Nash | 48.84±0.21 | 46.23±0.17 | 50.07±0.19 | 46.50±0.22 | 35.87±0.20 | 26.04±0.16 | 40.52±0.25 | 27.11±0.13 |
| Share-FAMO | 48.92±0.17 | 46.11±0.10 | 50.19±0.09 | 46.78±0.13 | 35.97±0.14 | 26.17±0.09 | 40.71±0.11 | 27.35±0.18 |
| MMOE-Fix | 47.74±0.09 | 44.01±0.05 | 48.61±0.11 | 44.23±0.04 | 35.29±0.16 | 25.67±0.09 | 40.04±0.26 | 26.87±0.11 |
| MMOE-DWA | 47.78±0.40 | 44.57±0.13 | 48.44±0.24 | 44.79±0.09 | 35.68±0.46 | 26.13±0.29 | 40.22±0.57 | 27.28±0.32 |
| MMOE-PE | 46.58±0.22 | 43.72±0.15 | 47.37±0.11 | 43.94±0.14 | 35.39±0.27 | 26.19±0.11 | 39.78±0.39 | 27.31±0.14 |
| MMOE-Nash | 47.88±0.27 | 46.49±0.29 | 51.17±0.22 | 47.10±0.31 | 35.79±0.18 | 26.02±0.27 | 41.03±0.20 | 26.39±0.28 |
| MMOE-FAMO | 47.93±0.32 | 46.42±0.23 | 51.24±0.19 | 47.15±0.21 | 35.92±0.21 | 26.14±0.17 | 41.15±0.39 | 26.62±0.14 |
| RMTL | 62.84±0.24 | 49.92±0.20 | 67.04±0.15 | 50.89±0.25 | 33.95±0.35 | 24.23±0.41 | 39.87±0.32 | 25.27±0.29 |
| PMORS | 63.14±0.15 | 51.02±0.17 | 67.45±0.20 | 52.07±0.13 | 34.16±0.26 | 24.09±0.23 | 39.86±0.19 | 25.31±0.27 |
| MOPRL | 61.18±0.19 | 50.74±0.10 | 64.76±0.25 | 51.65±0.02 | 33.99±0.11 | 24.31±0.08 | 38.98±0.08 | 25.57±0.08 |
| MOGCSL | **65.43±0.15** | **52.92±0.11** | **69.28±0.14** | **53.90±0.14** | **36.30±0.25** | 25.24±0.15 | **41.92±0.35** | 26.67±0.19 |

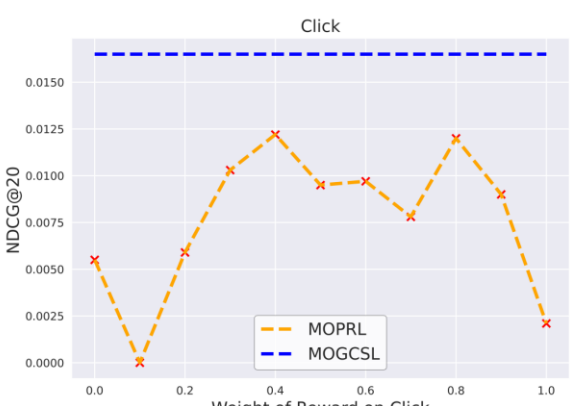
- Comparison between MOGCSL and MOPRLs with different weight combinations









- Complexity Comparison

- MOGCSL significantly benefits from a smaller model size and faster training speed.

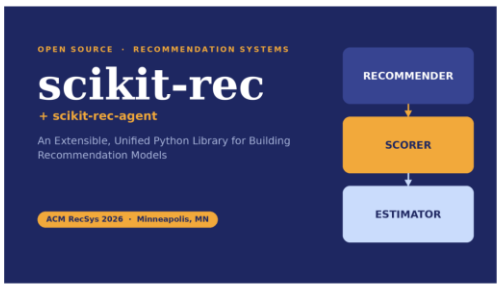
| | Model Size | Training time |
|---|---|---|
| Share-Fix | 14.0M | 9.6Ks |
| Share-DWA | 14.0M | 5.3Ks |
| Share-PE | 14.0M | 5.6Ks |
| Share-Nash | 14.0M | 8.6Ks |
| Share-FAMO | 14.0M | 5.1Ks |
| MMOE-Fix | 14.1M | 10.2Ks |
| MMOE-DWA | 14.1M | 9.5Ks |
| MMOE-PE | 14.1M | 60.5Ks |
| MMOE-Nash | 14.1M | 12.4Ks |
| MMOE-FAMO | 14.1M | 8.8Ks |
| RMTL | 17.5M | 100.2Ks |
| PMORS | 14.2M | 10.4Ks |
| MOPRL | 9.1M | 3.2Ks |
| MOGCSL | 9.1M | 3.0Ks |

- Implementation on Intuit's front-page recommendation

  - Incorporate long-term reward (login and/or revenue) as goals in GCSL

| | GCSL-L | GCSL-LR | Base |
|---|---|---|---|
| Average Precision | 0.4003 | 0.3990 | 0.3003 |
| Log-loss | 0.05727 | 0.05714 | 0.0664 |

- Intuit recently released a public version of their internal recommender system library, and I implemented the MOGCSL method in it during my summer internship at Intuit.




**Introducing Scikit-Rec: An Extensible and Unified Python Library for Recommendation Models — with Scikit-Rec-Agent, a Conversational Interface**



**Shankar Sankararaman** ✓
Principal Scientist and AI/ML Architect

September 28, 2026

**Scaling Intuit's Recommendation Models Across Multiple Products and Use Cases**

Intuit's portfolio — TurboTax, QuickBooks, Credit Karma, Mailchimp — creates a genuinely wide range of personalization opportunities: surfacing the next best action, personalizing a homepage for a first-time visitor, deciding who to offer an upsell to, or figuring out which deductible expenses to highlight for a self-employed filer. These aren't variations on one problem — some call for ranking, some for real-time exploration under uncertainty, some for estimating the causal effect of an offer, some for reasoning over a user's sequence of actions over time. And critically: some of these personalization efforts are tightly connected across products (shared users, shared signals), while others are completely independent.

architecture. **Shijun Li**, an Intuit intern and Ph.D. candidate under Prof. **Joydeep Ghosh**, implemented the GCSL methodology and demonstrated the library's scalability. **Emile Contal** and **Derek Tingle** of Crossing Minds (now part of OpenAI) provided valuable architectural and strategic guidance along the way. In addition, Intuit's leadership — **Nhung Ho**, **Kumar Sricharan**, **Alpesh Gaglani**, and **Zhewen (Joanne) Fan, ph.D** — provided sustained support throughout the development of these libraries during the past three years.

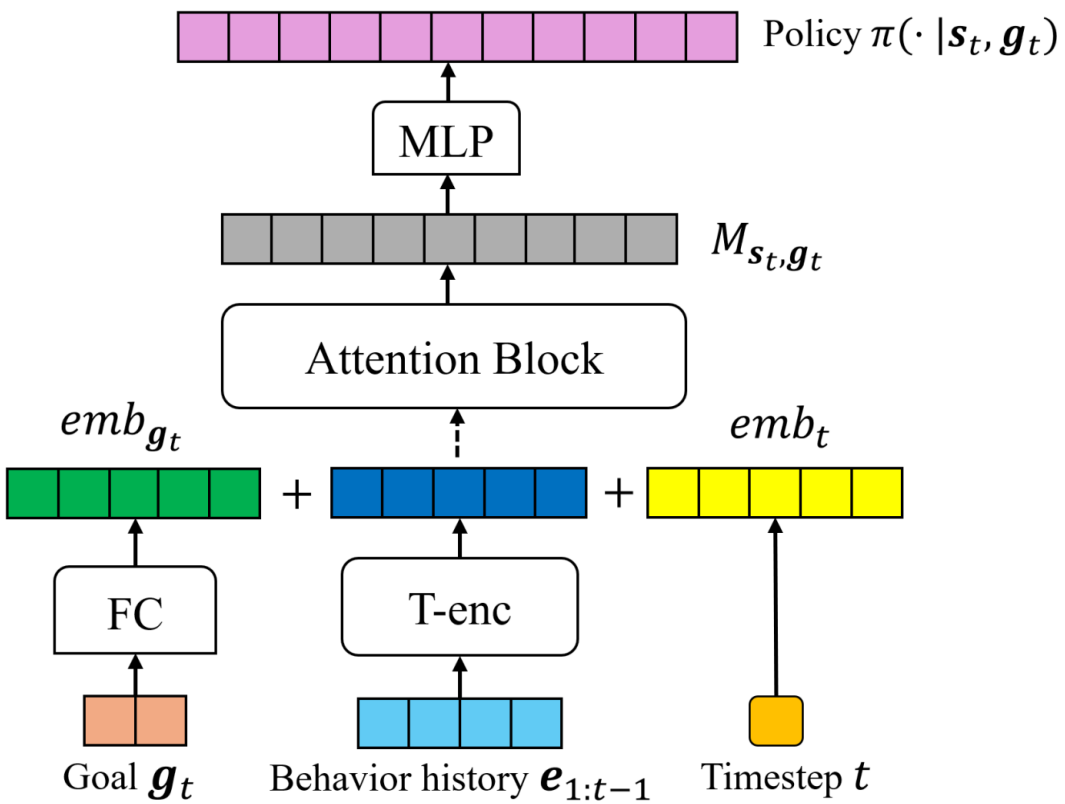
Both scikit-rec and scikit-rec-agent are open source under Apache 2.0, with public notebooks so the results can be reproduced independently. Both industry practitioners and researchers working on recommendation problems are welcome to use it. The scikit-rec library provides an organic approach to train and evaluate multiple types of recommendation models under the same architecture. Researchers and innovators developing new methods are especially encouraged to contribute and expand the libraries to include new methods, considering the library provides an easy way to benchmark and compare new methods against existing ones.

**Links & References**

[1] scikit-rec (GitHub repository). **https://github.com/intuit/scikit-rec**

[2] Shijun Li, Hilaf Hasson, Jing Hu, and Joydeep Ghosh. 2024. Goal-Conditioned Supervised Learning for Multi-Objective Recommendation. arXiv:2412.08911. **https://arxiv.org/abs/2412.08911**

- Structure of MOGCSL



- Goal selection criteria in our experiments: ensures that no goal within the achievable set is superior to the selected goal across all objectives

$$\tilde{\boldsymbol{g}}_b^a = \tilde{\boldsymbol{g}} \in \boldsymbol{G}^a, \text{ s.t. } \nexists \tilde{\boldsymbol{g}}' \in \boldsymbol{G}^a \setminus \tilde{\boldsymbol{g}}, \ \tilde{g}'_i \geq \tilde{g}_i \ \forall i \in [1, d].$$

- Comparison of statistical strategy and CVAE-based goal-choosing strategies with MOGCSL.

| [RetailRocket] | Purchase (%) | | | | Click (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | HR@10 | NG@10 | HR@20 | NG@20 | HR@10 | NG@10 | HR@20 | NG@20 |
| MOGCSL-S | **65.43**$\pm$**0.15** | **52.92**$\pm$**0.11** | 69.28$\pm$0.14 | 53.90$\pm$0.14 | 36.30$\pm$0.25 | 25.24$\pm$0.15 | 41.92$\pm$0.35 | 26.67$\pm$0.19 |
| MOGCSL-C | 65.01$\pm$0.07 | 52.89$\pm$0.04 | **69.34**$\pm$**0.05** | **54.00**$\pm$**0.04** | **36.54**$\pm$**0.02** | **25.41**$\pm$**0.04** | **42.20**$\pm$**0.03** | **26.84**$\pm$**0.06** |

| [Tenrec] | Like (%) | | | | Click (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | HR@10 | NG@10 | HR@20 | NG@20 | HR@10 | NG@10 | HR@20 | NG@20 |
| MOGCSL-S | 5.96$\pm$0.17 | 2.15$\pm$0.12 | 6.93$\pm$0.20 | 2.70$\pm$0.11 | 4.87$\pm$0.13 | 1.52$\pm$0.08 | 5.67$\pm$0.14 | 1.95$\pm$0.11 |
| MOGCSL-C | **6.78**$\pm$**0.07** | **2.99**$\pm$**0.02** | **7.84**$\pm$**0.05** | **3.86**$\pm$**0.04** | **5.66**$\pm$**0.05** | **2.14**$\pm$**0.05** | **6.73**$\pm$**0.03** | **2.68**$\pm$**0.07** |

# Outline

- RL on CRS
- Trustworthy Evaluation of RS
- Offline RL
- Multi-Agent System Optimization
- Current Projects/Interests

- Training Algorithm

**Algorithm 1:** Training of MOGCSL

**Input:** training data $\mathcal{D}_{tr}$, batch size $B$, model parameters $\theta$
**Intialization:** initialize parameters $\theta$
Relabel all the rewards with goals according to Eq. (1)
**repeat**
    Randomly sample a batch of $(\boldsymbol{s}_t, a_t, \boldsymbol{g}_t)$ from $\mathcal{D}_{tr}$
    Compute the representation $\boldsymbol{M}_{\boldsymbol{s}_t, \boldsymbol{g}_t}$ via Eq. (2)-Eq.(3)
    Derive the prediction logits via Eq. (4)
    Calculate the loss function $\mathcal{L}(\theta)$ via Eq. (5)
    Update $\theta$ by minimizing $\mathcal{L}(\theta)$ with stochastic gradient descent: $\theta \leftarrow \theta - \eta \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$
**until** *convergence*

- Inference Algorithm

**Algorithm 2:** Inference of MOGCSL

**Input:** state $\boldsymbol{s}'$, sample size $K$, policy model $\pi$, utility principle $U$, prior $q(\boldsymbol{g}'|\boldsymbol{s}')$, distribution of achievable goals $P(\boldsymbol{g}^a|\boldsymbol{s}', \boldsymbol{g}', \pi)$

**Intialization:** set of potential input goals $\boldsymbol{G}' = \emptyset$, set of expected achievable goals $\boldsymbol{G}^a = \emptyset$

**for** $k = 1, \ldots, K$ **do**

  Sample a $\boldsymbol{g}'_k$ from $q(\boldsymbol{g}'|\boldsymbol{s}')$

  Compute the expectation of the achievable goal through sampling: $\tilde{\boldsymbol{g}}^a_k = \mathbb{E}_{\boldsymbol{g}^a_k \sim P(\cdot|\boldsymbol{s}', \boldsymbol{g}'_k, \pi)} \boldsymbol{g}^a_k$

  $\boldsymbol{G}' = \boldsymbol{G}' \cup \boldsymbol{g}'_k$

  $\boldsymbol{G}^a = \boldsymbol{G}^a \cup \tilde{\boldsymbol{g}}^a_k$

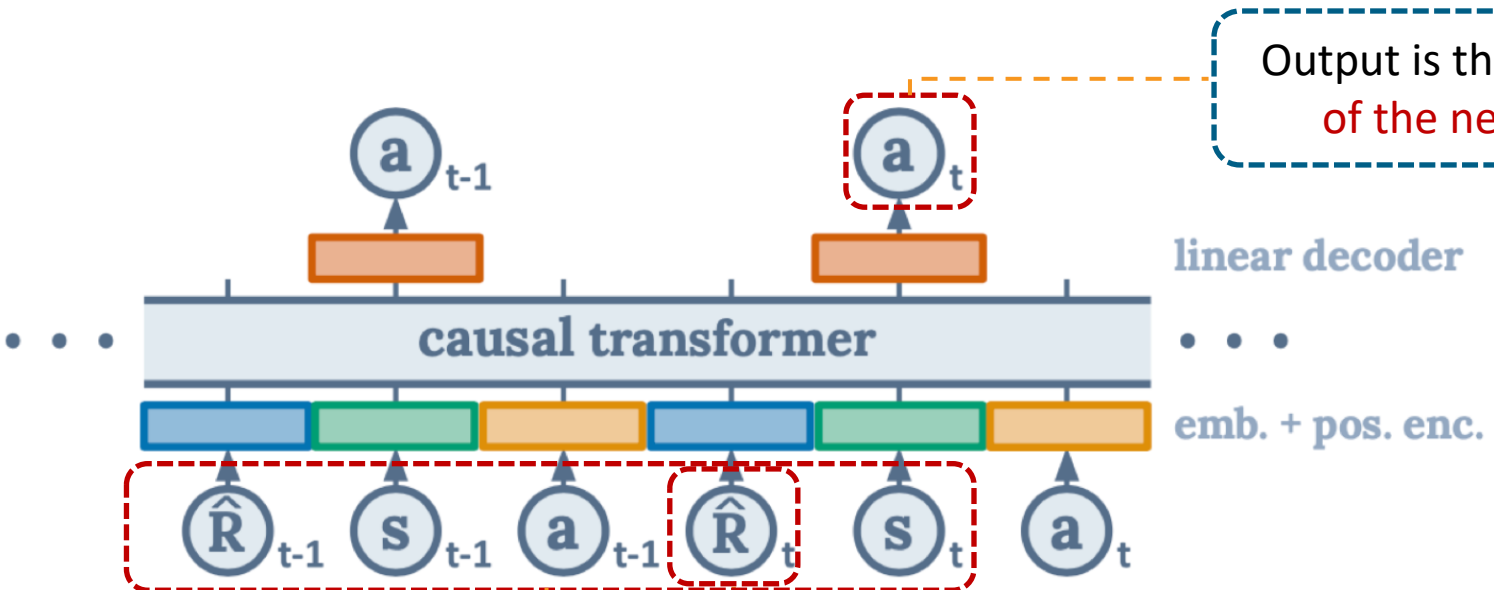
Choose the best $\tilde{\boldsymbol{g}}^a_b$ from $\boldsymbol{G}^a$ according to $U(\tilde{\boldsymbol{g}}^a)$

Choose corresponding $\boldsymbol{g}'_b$ from $\boldsymbol{G}'$

**Return:** $\pi(\cdot|\boldsymbol{s}', \boldsymbol{g}'_b)$

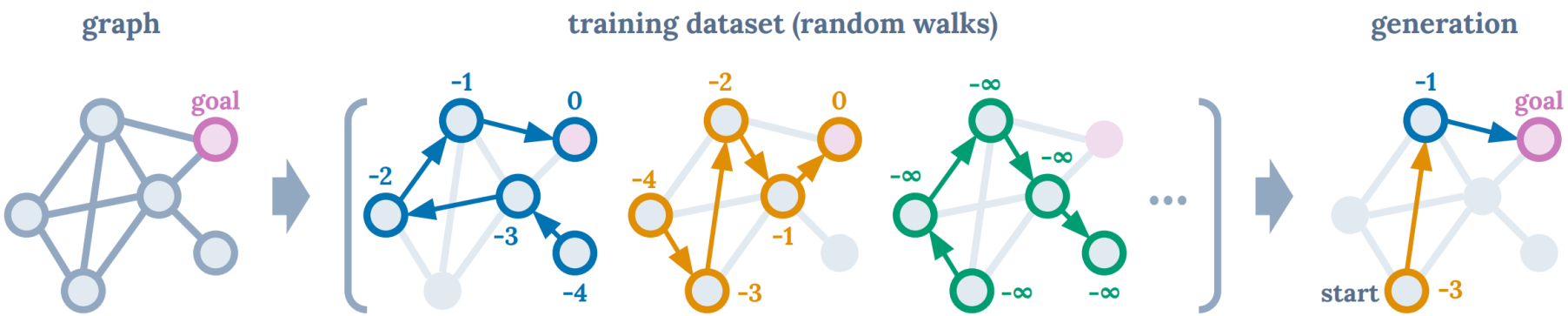
- What do goals mean?
  - One straightforward definition:
    $g = R_t = \sum_{t'=t}^{T} [\gamma^{t'} * r_{t'}]$, which is the target cumulative reward
  - Other forms: desired property/feature vector, desired final state, language goals, image goals, etc.
- How to achieve the goals?
  - Good to be trained in a supervised, goal-conditioned framework with offline sequential data.

- Example: Decision Transformer*




*: Chen, Lili, et al. "Decision transformer: Reinforcement learning via sequence modeling." NeurIPS 2021.

- Decision Transformer finds the shortest path

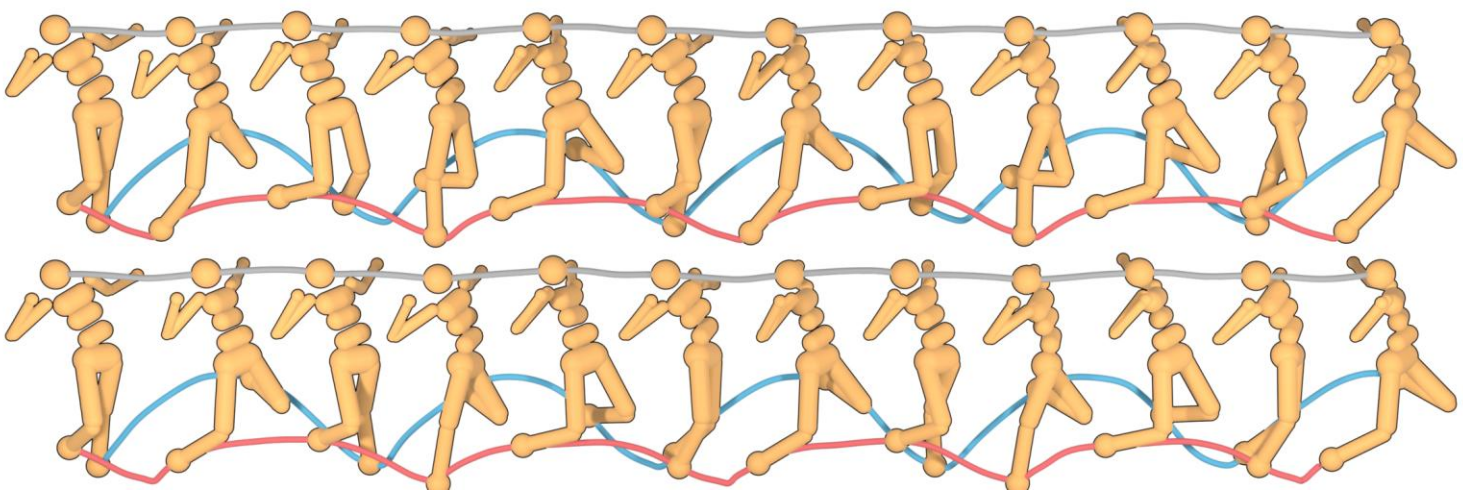



- Trained on random walk sequences and per-node returns-to-go.
  Given an initial state and a large return, it sequences optimal paths.

- A talk about Decision Transformer: YouTube Link.

- Difference from imitation learning

  - Imitation Learning: given expert sequential data, then imitate the behaviors of expert.
    Assumption: training data is optimal!



- Goal-conditioned RL: being able to learn from data generated by sub-optimal policies. And it may perform a better policy in inference stage.

- Why?
  Instead of learning from a certain policy, goal-conditioned RL learns to model the projection: $S \times G \rightarrow A$

- Every trajectory is a successful demonstration for the goal that it actually reaches, even if the policy is sub-optimal for the initially set goal.