COMBOU: Leveraging Unlabelled Data in Conservative Offline Model-Based RL

Frank Zhao[1], Tommy Li[1]

Department of Computer Science, Stanford University

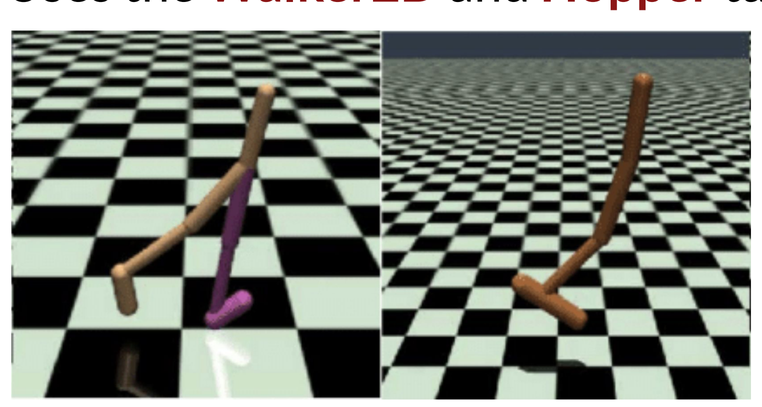
Stanford Computer Science

Project Overview

- Offline RL learns from static datasets that require **reward annotation**.
- In many cases, labelling reward is **costly**.
- Common to have a small amount of **labelled task-specific data** and a large amount of **unlabelled task-agnostic data** (state, action, next_state) without reward.
- **Problem Statement:** leverage the use of unlabelled data in offline model-based RL, specifically in Conservative Offline Model Based Policy Optimization (COMBO).
- **Previous Literature:** in model-free methods, reward prediction performs poorly, setting all unlabelled data's reward to 0 (**UDS**) is effective.
- **COMBO** consists of 3 parts:
  - Dynamics (state & reward) Training
  - Critics Training
  - Conservative Policy Evaluation
- This project explores how to incorporate unlabeled data into these three parts.
- **Main Results:** UDS and reward prediction with built-in pessimism both work very well (**~30%** improvement from baseline COMBO method)!

Datasets & Metrics

- D4RL Benchmark for Offline RL.
- Uses the **Walker2D** and **Hopper** tasks.



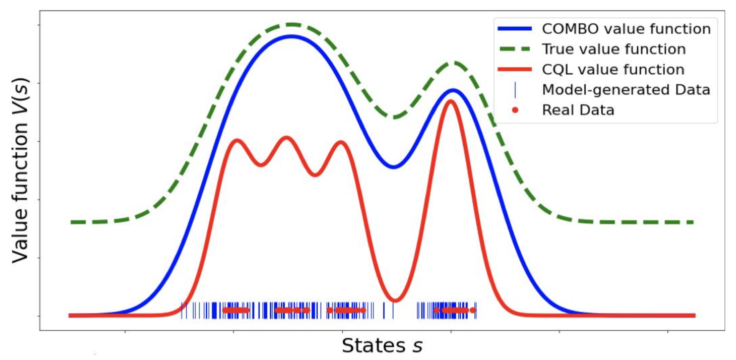
- **10k labelled expert samples** (s, a, s', r) ~ **L** from a policy trained with SAC + **1M random unlabelled samples** (s, a, s') ~ **U** from a random policy.
- Resembles the case unlabelled data is of low quality and even irrelevant to the target task.
- **Metric:** Average normalized evaluation episode reward in the last 100 training epochs.
- We use the **Adam** optimizer. All model backbones are **MLPs** that follow the COMBO paper.

Methods & Experiments

- **COMBO** in detail, given **labeled data L** and **policy** π:
  1. Train **dynamics model** $T_\theta(s', r|s, a)$ on **L**
  2. Iterate:
     a) Rollout dynamics model for **model data M**
     b) **Conservatively evaluate** critics:

$$Q^\pi := \quad argmin_Q \mathbb{E}_{(s,a,s',r)\sim L\cup M}[(Q(s,a) - (r + \gamma\mathbb{E}[Q(s',a')]))^2] + \alpha\mathbb{E}_{(s,a)\sim M}[Q(s,a)] - \alpha\mathbb{E}_{(s,a)\sim L}[Q(s,a)]$$

     c) Improve policy π based on updated critics



- **Baseline 1 (COMBO with no data sharing):** Run COMBO on 10k expert labelled data **L** only.
- **Baseline 2 (naive reward prediction):**
  - Use **L** and **U** to train dynamics model to predict next state (s' | s, a).
  - Use **L** alone to train a reward model **R** and use **R** to fill in the rewards for data in **U**.
  - Run COMBO on **L** and **U**.
- **Variant 1 (only use unlabelled data for training state dynamics):**
  - Use **L** and **U** to train dynamics model to predict next state (s' | s, a).
  - Use **L** alone to train a reward model **R**. Run COMBO on **L** alone.
- **Variant 2 (UDS):**
  - Set the reward of all unlabelled data in **U** to 0. Then combine **L** & **U** to run COMBO on them.
- **Variant 3 (reward prediction with built-in pessimism):**
  - Use **L** and **U** to train dynamics model to predict next state (s' | s, a).
  - Use **L** alone to train a reward model **R** and use **R** to fill in the rewards for data in **U**.
  - Run COMBO on **L** and **U** with **built-in pessimism** on **U** (step 3 second line): $\mathbb{E}_{(s,a)\sim M\cup U}$

Discussions & Future Research

Discussions:

- **Reward prediction with built-in pessimism** is very effective for leveraging unlabelled data!
- COMBO archives 103.3 using 2M medium-expert data on Walker2D. CQL+UDS achieves 81.5 in the same setup on Hopper. → **Our method is potentially superior!**
- Using unlabelled data to train state dynamics is useful. Naive reward prediction doesn't work.
- Variant 2 (UDS) doesn't need built-in pessimism as we already assign lowest reward to unlabelled data. Its performance is more variable.

Future Research:

- Needs further investigation on overfitting in Hopper environment.
- Test on more environments to figure out the methods' strength and weaknesses.
- Incorporate CDS into the framework.
- Experiment with multitask learning environments as this approach naturally applies.

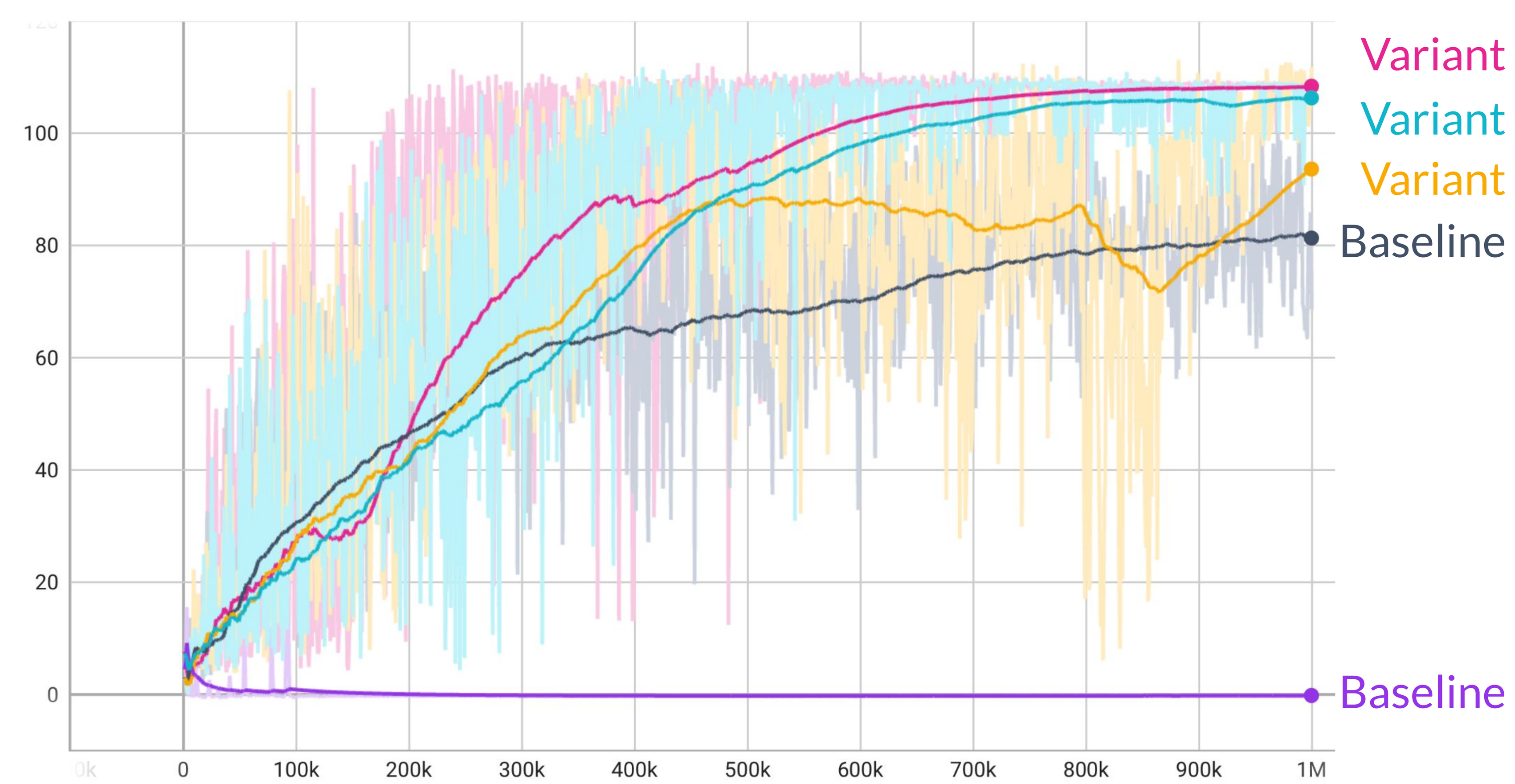
References

[1] T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn, 'COMBO: Conservative Offline Model-Based Policy Optimization', *arXiv [cs.LG]*. 2022.

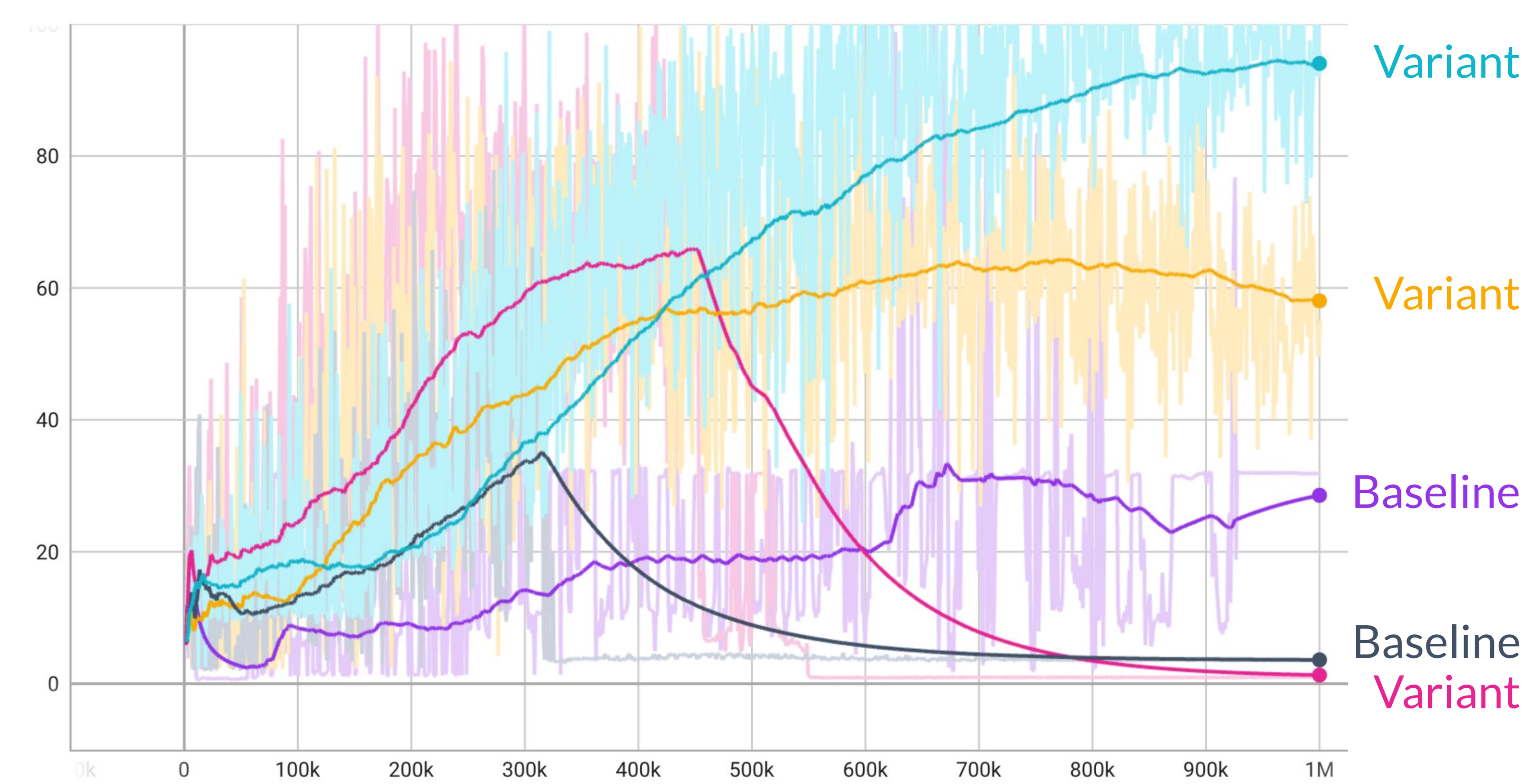
[2] T. Yu, A. Kumar, Y. Chebotar, K. Hausman, C. Finn, and S. Levine, 'How to Leverage Unlabeled Data in Offline Reinforcement Learning', *arXiv [cs.LG]*. 2022.

Results

Training Curves for Walker2D Environment:



Training Curves for Hopper Environment:



*Note that we discard the data after the sudden drop in baseline 1 and variant 2.

	Walker2D		Hopper	
	Avg. Eval Reward	Stdev. Eval Reward	Avg. Eval Reward	Stdev. Eval Reward
COMBO (Baseline 1)	82.372	10.187	39.918	16.656
Reward Pred. (B2)	-0.167	0.017	29.653	9.960
Variant 1	100.794	9.877	55.355	9.292
Variant 2	**108.534**	1.611	66.466	16.027
Variant 3	106.064	5.444	**95.167**	8.381