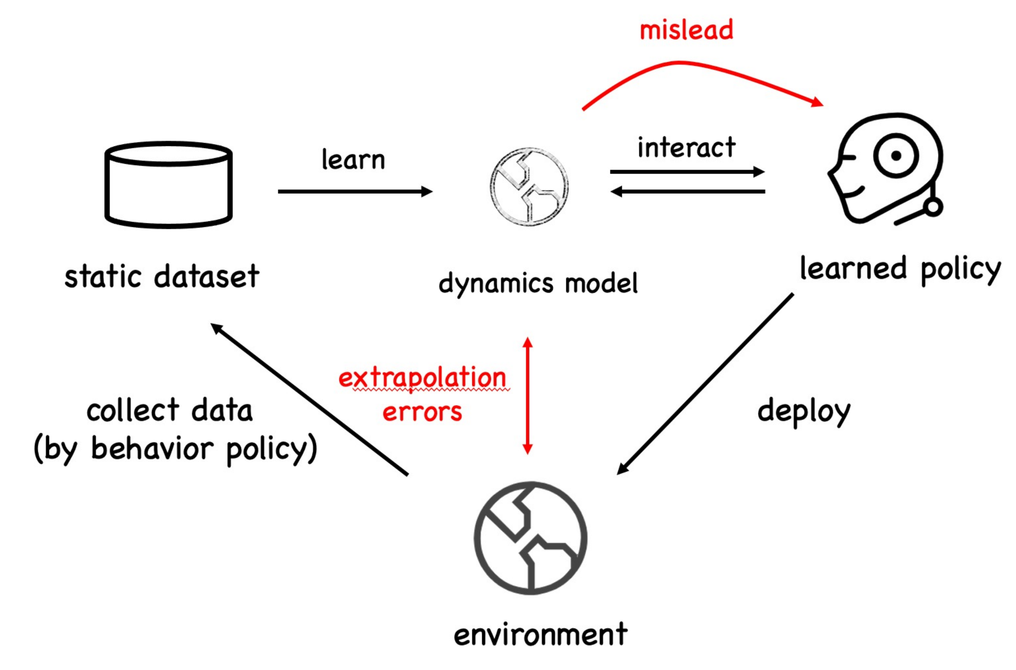
MAPLE: Offline Model-based Adaptable Policy Learning

Xiong-Hui Chen[1], Yang Yu[1,3,*], Qingyang Li[2], Fan-Ming Luo[1], Zhiwei Qin[2], Wenjie Shang[2], Jieping Ye[2]
[1] National Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, China
[2] AI Labs, Didi Chuxing
[3] Polixir.ai

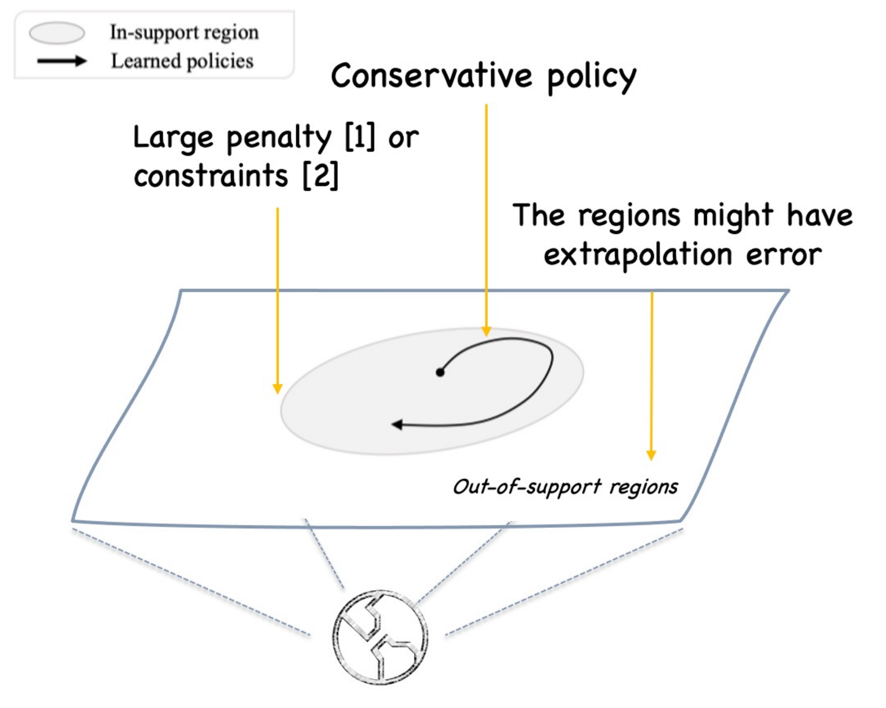
Background and Motivation

Challenges of Model-based Offline RL



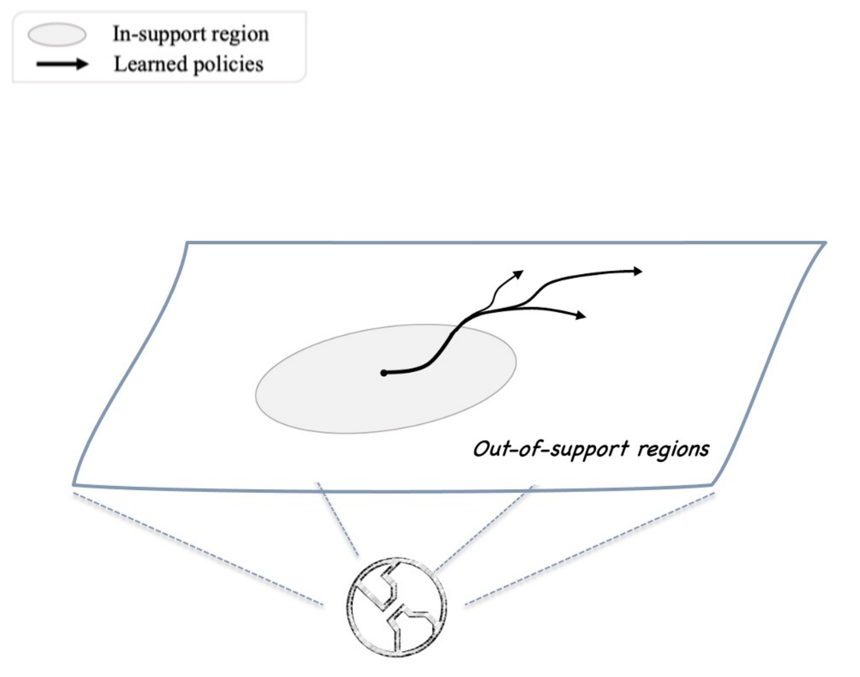
The extrapolation errors of learned dynamics models will mislead the direction of policy learning.

Model-based Offline RL via Conservatism



Conservatism guarantees the lower-bound performance of the learned policy, **but also limits the upper-bound performance.**

Our Research Question

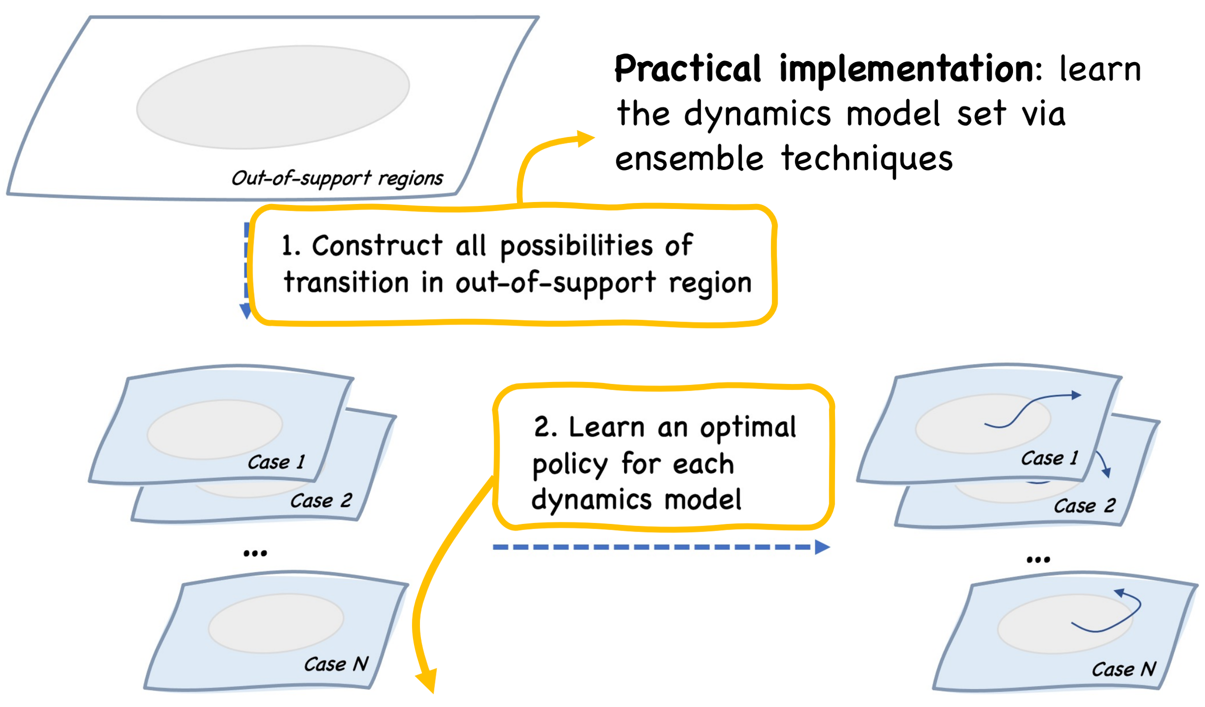


Can we the handle the decision-making problem in out-of-support regions directly?

Offline Model-based Adaptable Policy Learning

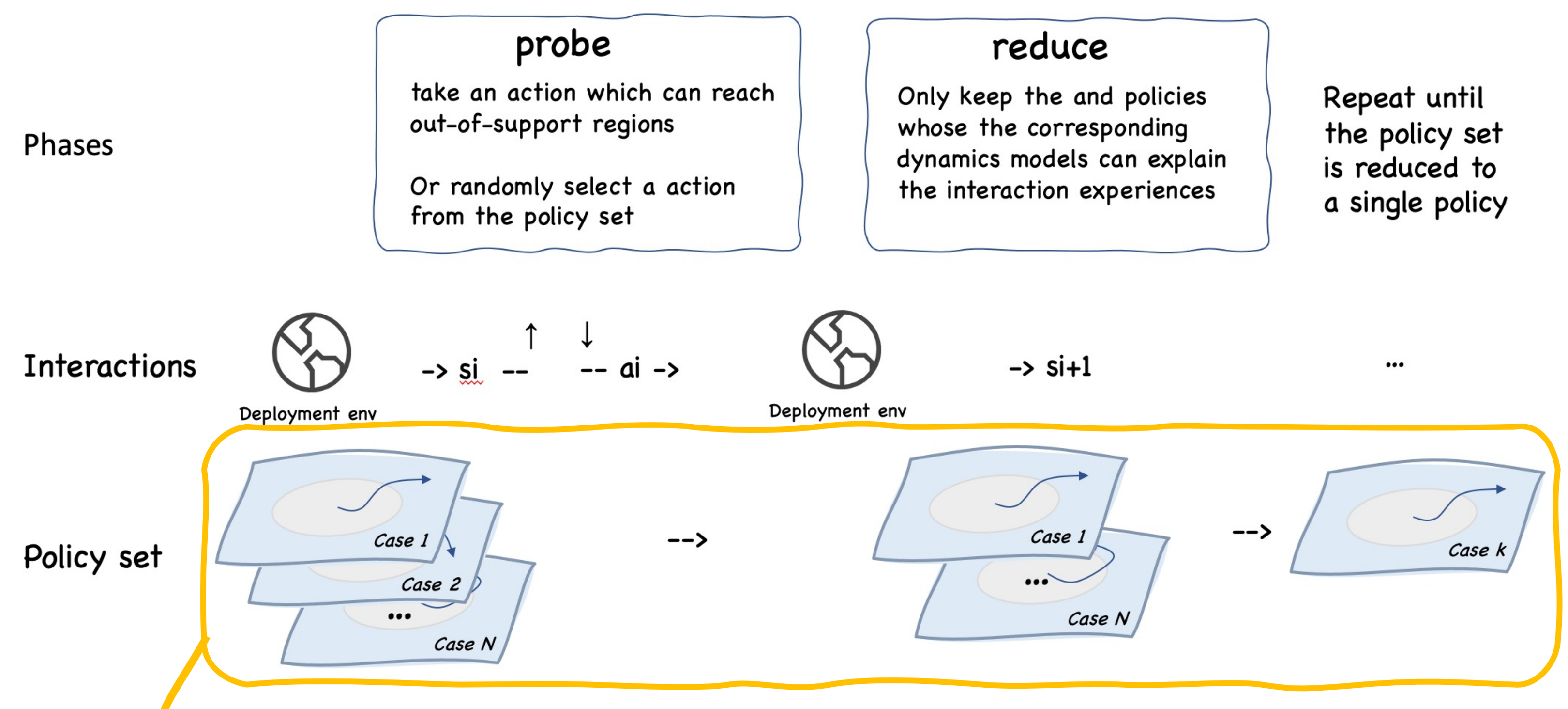
An ideal solution, named probe-reduce paradigm, and its practical implementation for decision-making in out-of-support regions

Training



Practical implementation:
Learn an adaptable meta-policy (based on an RNN context extractor).

Deployment



Practical implementation: feed the interaction trajectory $\tau_{0:t}$ in the deployment environment to the RNN context extractor and infer the environment-context for each timestep t. Repeat until the context is converged to z then we find the optimal policy $\pi(a|s,z)$

Experiment results and Discussion

Comparative Evaluation on Benchmark Tasks

We first test MAPLE in standard offline RL tasks with D4RL datasets [3].

Table 1: Results on MuJoCo tasks. Each number is the normalized score proposed by Fu et al. [30] of the policy at the last iteration of training, ± standard deviation. Among the offline RL methods, we bold the highest mean for each task.

Environment	Dataset	MAPLE	MOPO	MOPO-loose	SAC	BEAR	BC	BRAC-v	CQL
Walker2d	random	**21.7 ± 0.3**	13.6 ± 2.6	8.0 ± 5.4	4.1	6.7	9.8	0.5	7.0
Walker2d	medium	56.3 ± 10.6	11.8 ± 19.3	32.6 ± 18.0	0.9	33.2	6.6	**81.3**	79.2
Walker2d	mixed	**76.7 ± 3.8**	39.0 ± 9.6	35.7 ± 2.2	3.5	25.3	11.3	0.4	26.7
Walker2d	med-expert	73.8 ± 8.0	44.6 ± 12.9	66.7 ± 14.8	-0.1	26.0	6.4	66.6	**111.0**
HalfCheetah	random	**38.4 ± 1.3**	35.4 ± 1.5	35.4 ± 2.1	30.5	25.5	2.1	28.1	35.4
HalfCheetah	medium	**50.4 ± 1.9**	42.3 ± 1.6	44.0 ± 1.6	-4.3	38.6	36.1	45.5	44.4
HalfCheetah	mixed	**59.0 ± 0.6**	53.1 ± 2.0	36.9 ± 15.0	-2.4	36.2	38.4	45.9	46.2
HalfCheetah	med-expert	**63.5 ± 6.5**	63.3 ± 38.0	15.0 ± 6.0	1.8	51.7	35.8	45.3	62.4
Hopper	random	10.6 ± 0.1	11.7 ± 0.4	10.6 ± 0.6	11.3	9.5	1.6	**12.0**	10.8
Hopper	medium	21.1 ± 1.2	28.0 ± 12.4	16.9 ± 2.4	0.8	47.6	29.0	32.3	**58.0**
Hopper	mixed	**87.5 ± 10.8**	67.5 ± 24.7	83.1 ± 6.5	1.9	10.8	11.8	0.9	48.6
Hopper	med-expert	42.5 ± 4.1	23.7 ± 6.0	25.1 ± 1.8	1.6	4.0	111.9	0.8	**98.7**

The performance of MAPLE on 7 tasks is better than other SOTA algorithms. Besides, MAPLE reaches the best performance among the SOTA model-based conservative policy learning algorithms in 10 out of the 12 tasks.

Ability of adaptable policy in out-of-support regions

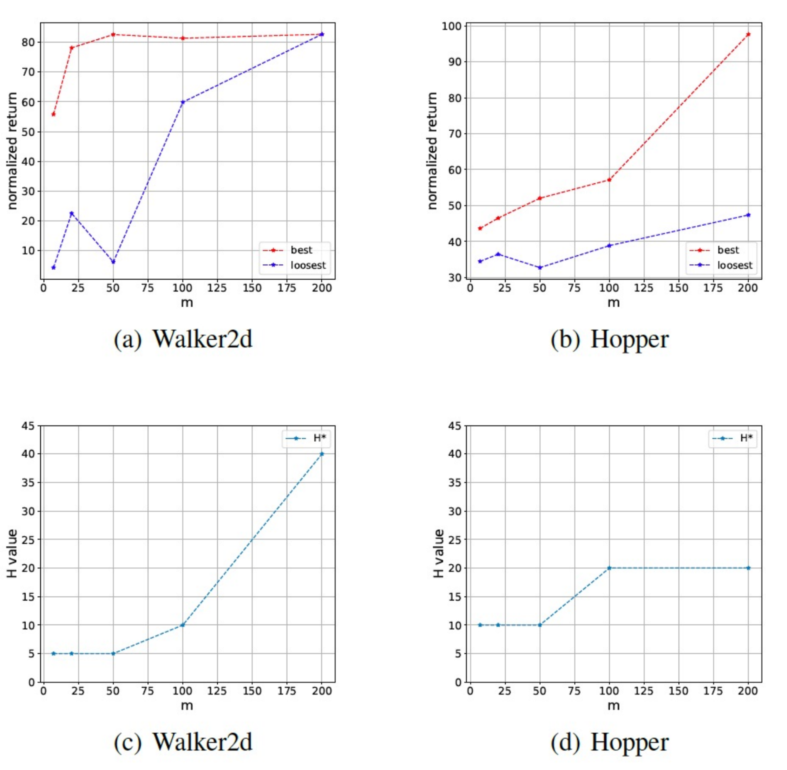


Figure 10: Illustration of hyper-parameters analysis on m. In the first row, we compare the normalized return of the best setting and the loosest setting. The x-axis is the model size m. For each m, the legend "best" is the setting that has the largest performance, among which model size is m. The legend "loosest" is the setting that $H = 40$. In the second row, we compare the best constraint setting for each model size m. For each m, the legend "H*" is the setting that H value of the best-performance setting among which model size is m.

Increase the model-set size is significantly helpful to find a better and robust adaptable policy via expanding the exploration boundary.

MAPLE with large dynamics model set

Table 2: Results on MuJoCo tasks with MAPLE-200.

Environment	Dataset	MAPLE-200	MAPLE
Walker2d	random	**22.1 ± 0.1**	21.7 ± 0.3
Walker2d	medium	**81.3 ± 0.1**	56.3 ± 10.6
Walker2d	mixed	75.4 ± 0.9	**76.7 ± 3.8**
Walker2d	med-expert	**107.0 ± 0.8**	73.8 ± 8.0
HalfCheetah	random	**41.5 ± 3.6**	38.4 ± 1.3
HalfCheetah	medium	48.5 ± 1.4	**50.4 ± 1.9**
HalfCheetah	mixed	**69.5 ± 0.2**	59.0 ± 0.6
HalfCheetah	med-expert	55.4 ± 3.2	**63.5 ± 6.5**
Hopper	random	**10.7 ± 0.2**	10.6 ± 0.1
Hopper	medium	**44.1 ± 2.6**	21.1 ± 1.2
Hopper	mixed	85.0 ± 1.0	**87.5 ± 10.8**
Hopper	med-expert	**95.3 ± 7.3**	42.5 ± 4.1

In all of the tasks, MAPLE-200 reaches at least similar performance to MAPLE. In the tasks like Walker2d-med-expert, HalfCheetah-mixed, Hopper-medium, and Hopper-med-expert, the performance improvement of MAPLE-200 is significant.

Conclusion and Take-home Messages

MAPLE gives another direction to handle the offline model-based learning problem: Learn to adapt in out-of-support regions.

Future work:

1. Generalization ability of the environment-context extractor with limited dynamics model.
2. Efficient/diverse dynamics model set generation process.

[1] Yu, Tianhe, et al. "Mopo: Model-based offline policy optimization." *arXiv preprint arXiv:2005.13239* (2020).
[2] Kidambi, Rahul, et al. "Morel: Model-based offline reinforcement learning." *arXiv preprint arXiv:2005.05951* (2020).
[3] Fu, Justin, et al. "D4rl: Datasets for deep data-driven reinforcement learning." *arXiv preprint arXiv:2004.07219* (2020).