Toward generalized context-based decision-making

Deep Demonstration Tracing: Learning Generalizable Imitator Policy for Runtime Imitation from a Single Demonstration;

Policy-conditioned Environment Models are More Generalizable.

Pre: Xiong-Hui Chen
LAMDA Group, Nanjing University
Superviosr: Yang Yu

南京大學 NANJING UNIVERSITY

LAMDA Learning And Mining from DatA

About

- 陈雄辉，南京大学20级硕转博博士生
- 南京大学LAMDA组俞扬老师指导下从事研究工作
- 研究关注：解决强化学习在交互成本敏感的真实应用场景中的挑战。目前的研究重点是离线强化学习、sim2real迁移、可泛化的真实世界环境模型学习。最近在探索基于大语言模型的决策和大型决策模型等相关课题。
- 10+篇论文发表在NeurIPS，ICML，ICLR，TPAMI等顶会上
- 关注学术成果转化：强化学习产品化落地：互联网企业（滴滴，美团），化工企业（施耐德），军工企业等
- 个人主页：https://xionghuichen.github.io/

Deep Demonstration Tracing: Generalizable Imitator Policy Learning

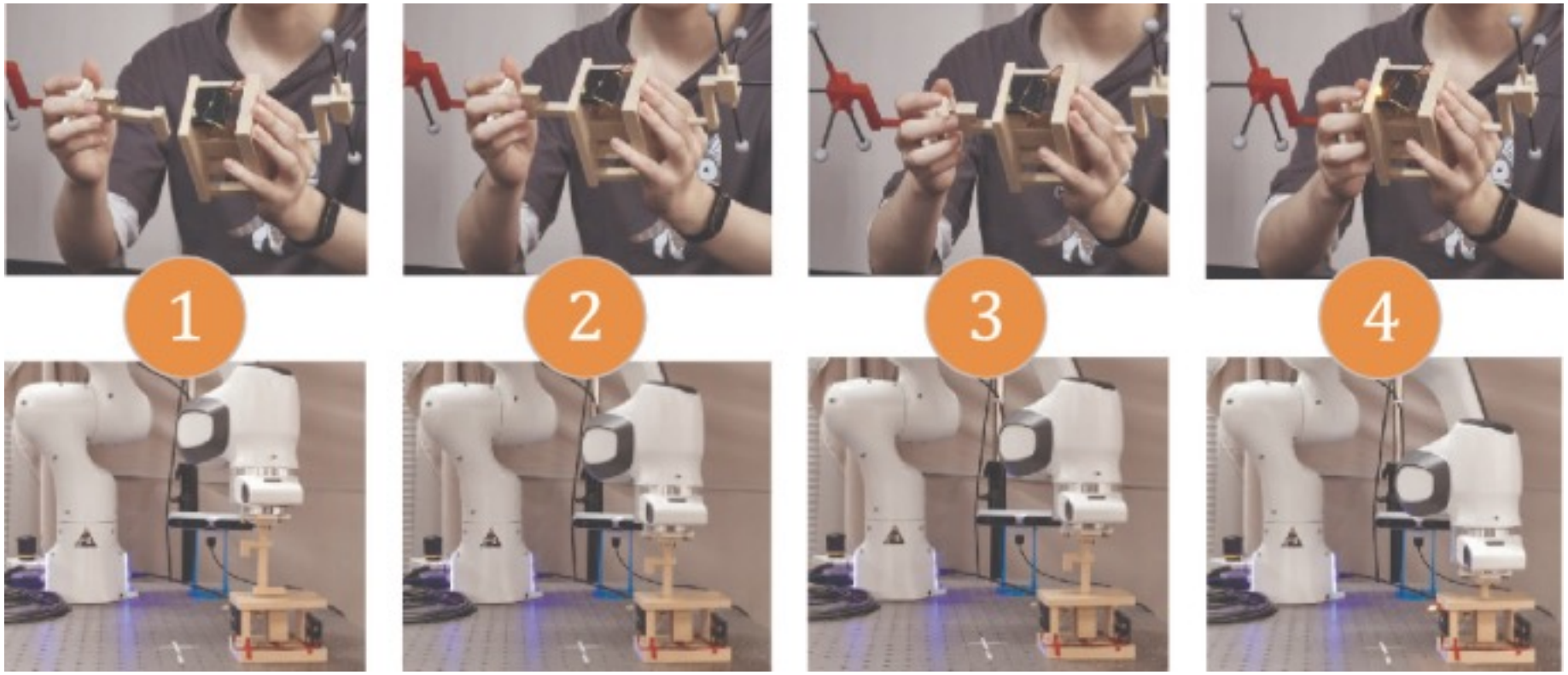
1. **Background**
2. Solution
3. Experiment
4. Take-home Messages

Deep Demonstration Tracing: Generalizable Imitator Policy Learning

Deep Demonstration Tracing: Learning Generalizable Imitator Policy for Runtime Imitation from a Single Demonstration

Xiong-Hui Chen [*1,2] Junyin Ye [*1,2] Hang Zhao [*3,2] Yi-Chen Li [1,2] Xu-Hui Liu [1,2] Haoran Shi [2] Yu-Yan Xu [2]
Zhihao Ye [1,2] Si-Hang Yang [1,2] Yang Yu [1,2] Anqi Huang [4,2] Kai Xu [3] Zongzhang Zhang [1]

The Vision of Runtime One-Shot Imitation Learning / Learning from a Single Demonstration



Runtime imitator policy: $\Pi(a|s,\tau)$, where $\tau \in \mathrm{T}$ is a unseen human demonstration.

Achieve any tasks directly "prompted" by corresponding demonstration τ.

A popular Paradigm: transformer with behavior cloning

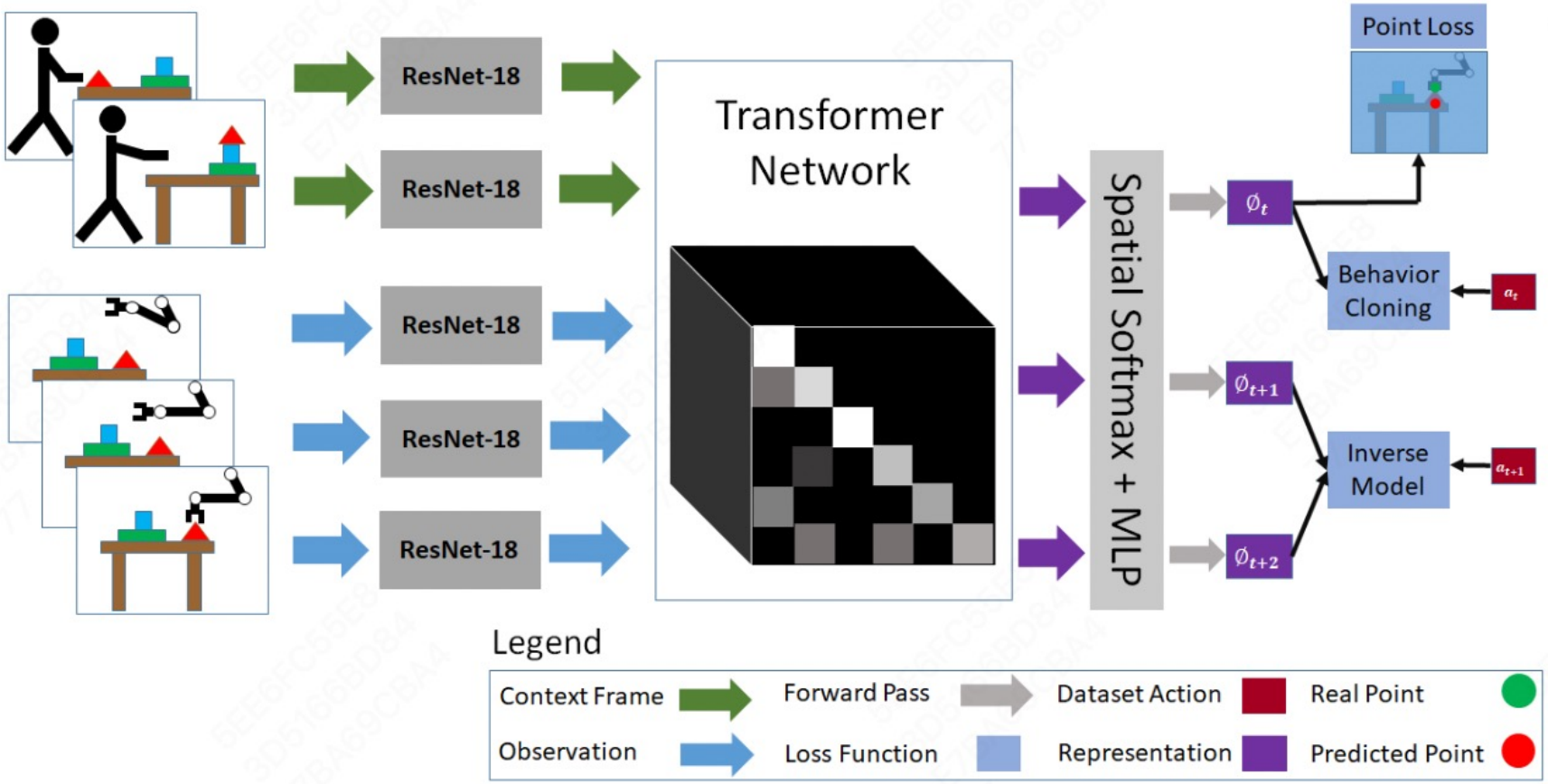
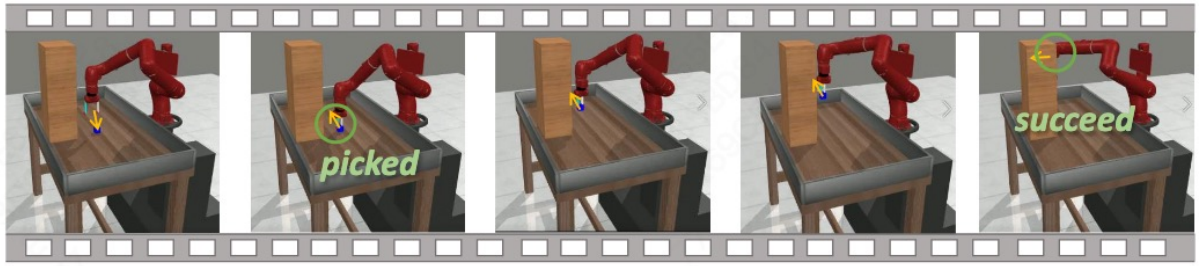
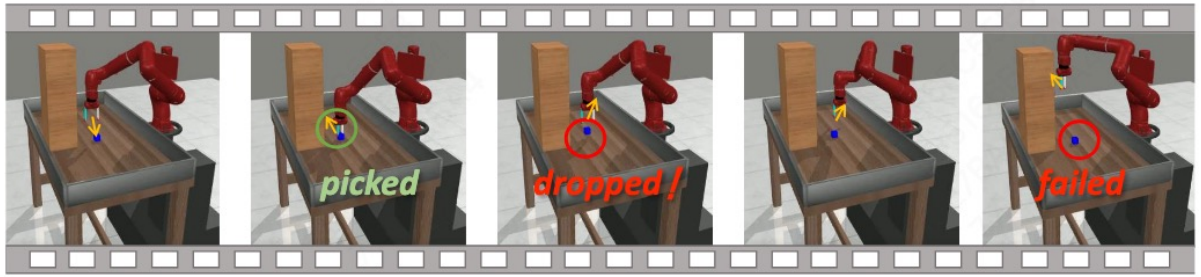


Figure 2: Our method uses a Transformer neural network to create task-specific representations, given context and observation features computed with ResNet-18 (w/ added positional encoding). The attention network is trained end-to-end with a behavior cloning loss, an inverse modelling loss, and an optional point loss supervising the robot's future pixel location in the image.

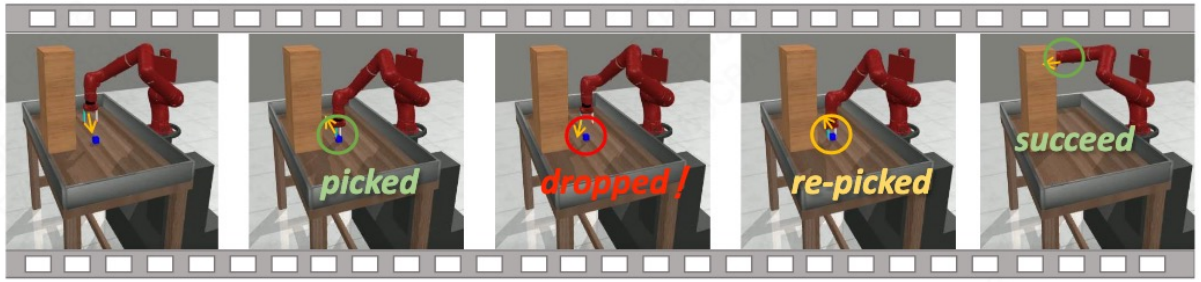
Generlization Challenge of Runtime One-Shot Imitation Learning (OSIL)



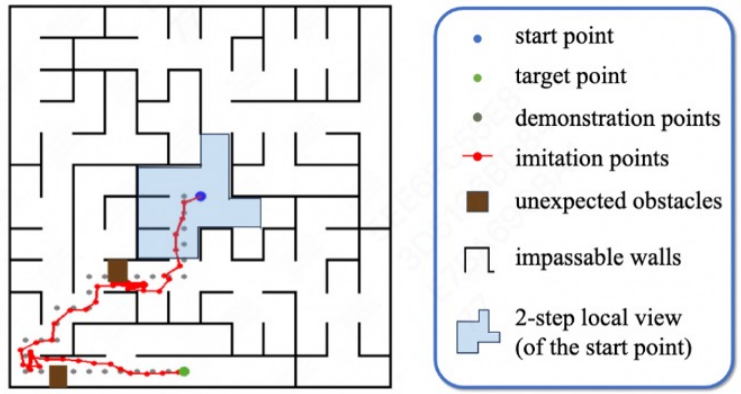
(a) Provided demonstration.



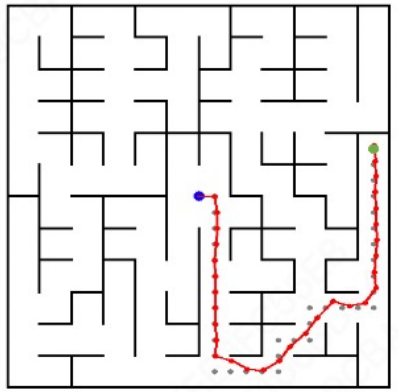
(b) Policy trained by a traditional OSIL method.



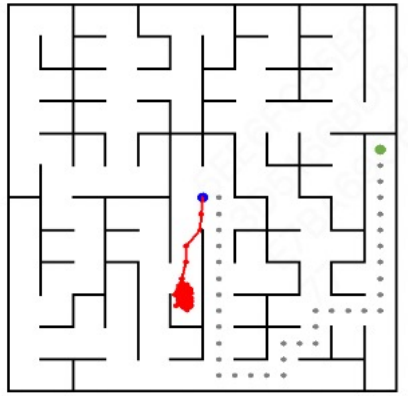
(c) Policy trained by DDT.



(A) Valet Parking Assist in Maze (VPAM)



Trained by DDT



Trained by traditional OSIL

- ***Unseen demonstrations in unseen envirnoments.***
  -> *incorrect representation caused by transformer.*
- ***Unforseen changes after demonstrations collection.***
  -> *limitation of behavior cloning.*

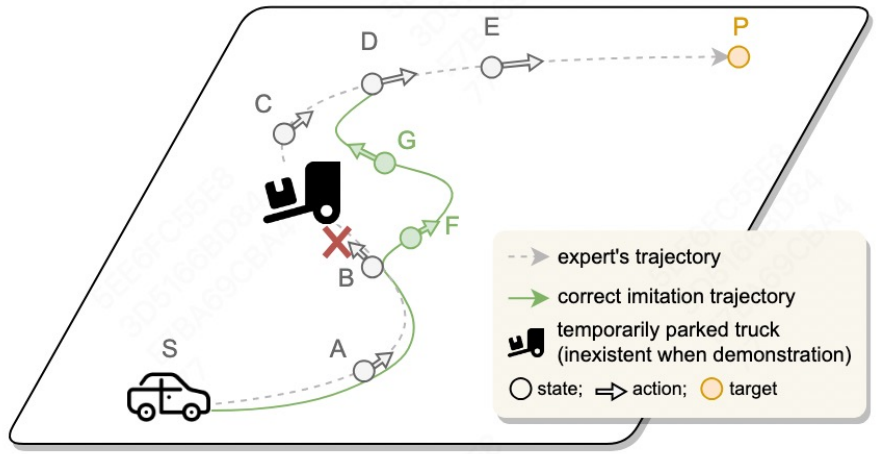
Deep Demonstration Tracing: Generalizable Imitator Policy Learning

1. Background

2. **Methodology**
    1. Demonstration transformer
    2. OSIL via meta-RL

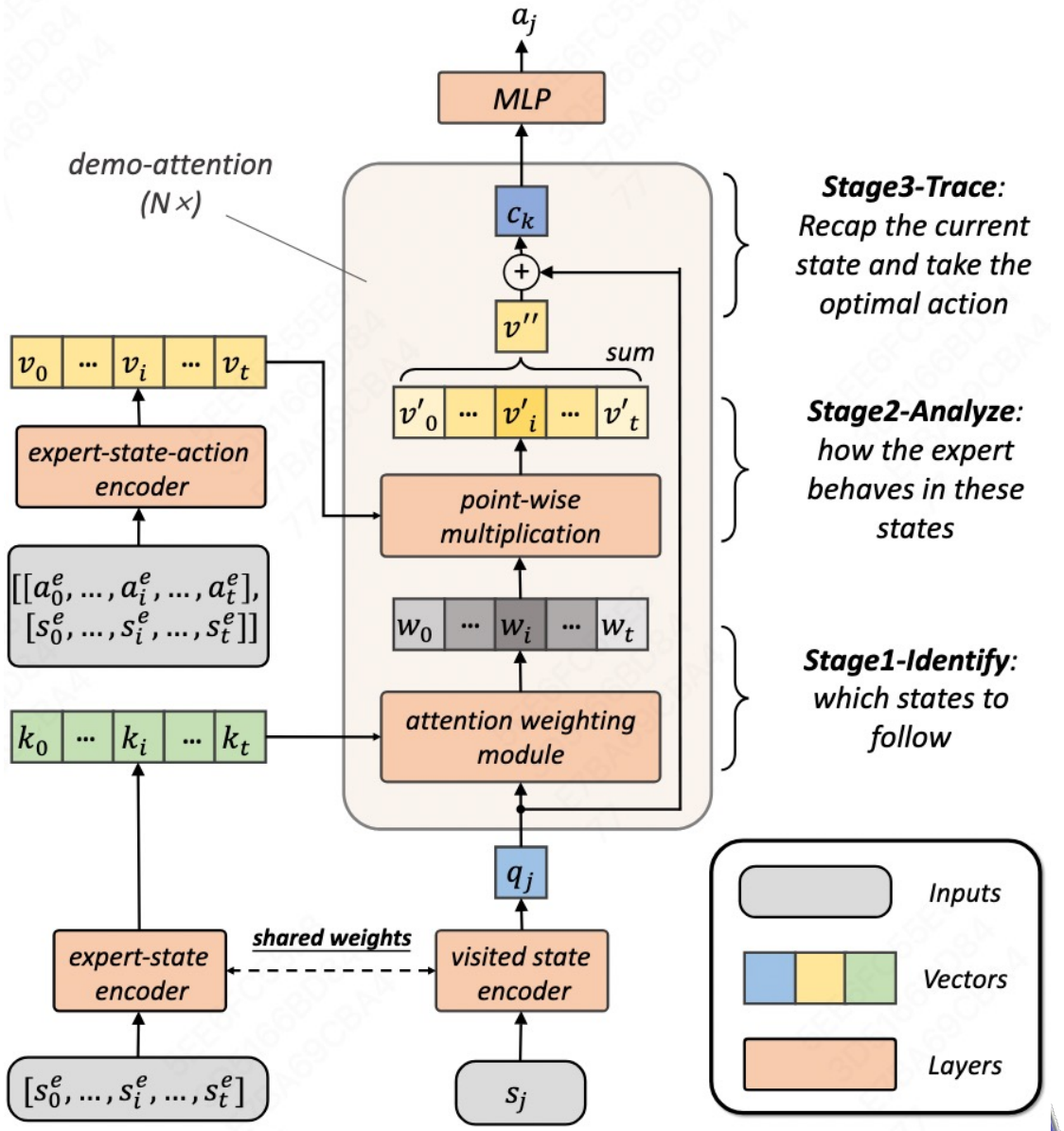
3. Experiment

Key1: Inject the inductive bias of "how human make decisions in runtime OSIL" into the imitator policy network



An example of 3-stage OSIL of humans.

- Stage 1: **Identify** relevant states within the trajectory based on the current state.
- Stage 2: **Analyze** the expert's behavior patterns associated with these states.
- Stage 3: **Trace** the expert's demonstrations based on the relationship between the current state and the expert's behavior patterns in the demonstrations.



Key2: Solve runtime one-shot imitation learning by context-based meta-RL, instead of supervised learning

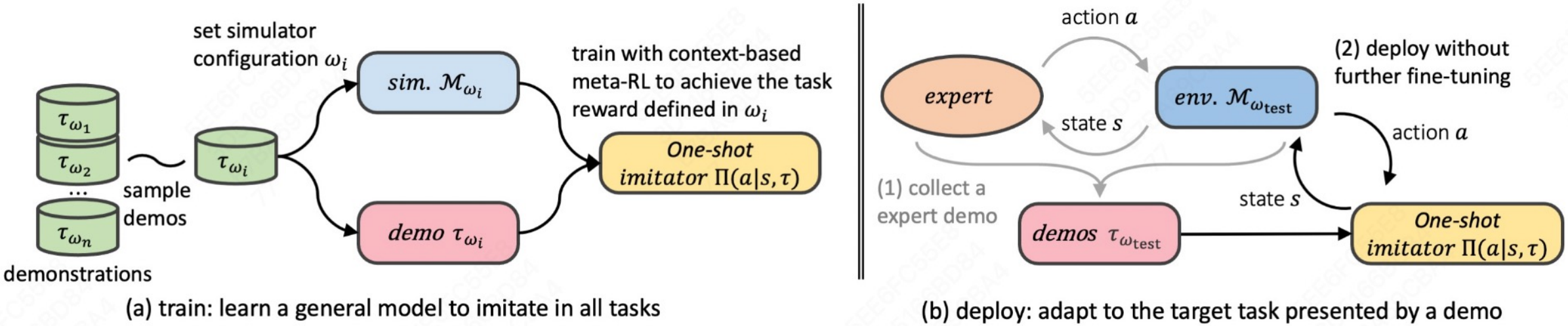


Illustration of the Training and Deploying Workflow for a Runtime One-shot imitator policy via context-based meta-RL.

- The unforeseen changes will randomly apprear in the simulators ($\mathcal{M}$).
- With meta-RL, the imitator policy will try to achieve *all of* the targets the same to the demonstration guided by 0-1 task rewards.
- In the process, the imitator policy will suffer from the unforseen changes and *have to handle them before achieve the targets*.

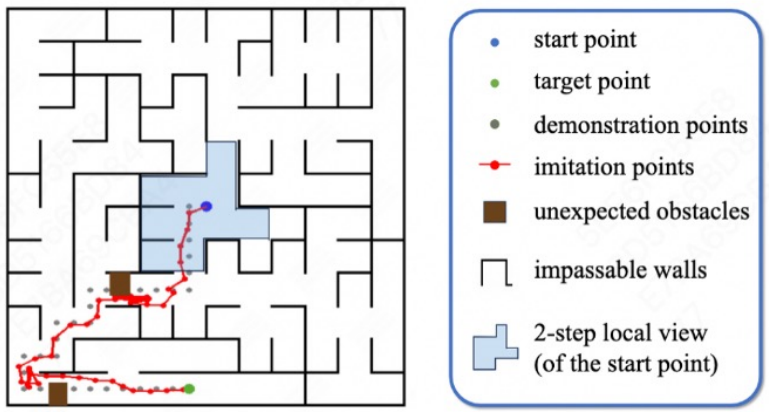
Deep Demonstration Tracing: Generalizable Imitator Policy Learning

1. Background
2. Methodology
3. **Experiment**

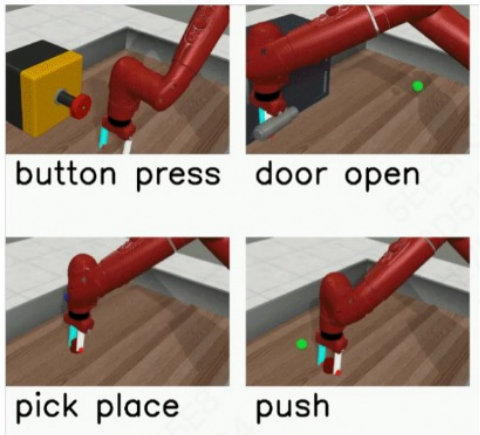
Research questions

1. **RQ1**: The one-shot imitation ability of DDT in unseen situations, including **unseen demonstrations, unseen environments, and unforeseen changes** after demonstration collection.
2. **RQ2**: Does demonstration transformer really imitating via tracing the demonstration?
3. **RQ3**: Can DDT have potential of performance improvement when scaling up the size of parameters and demonstration data, inspired by the **"Scaling Law"** in large language models.

Experiment: Valet Parking Assist in Maze



(A) Valet Parking Assist in Maze (VPAM)

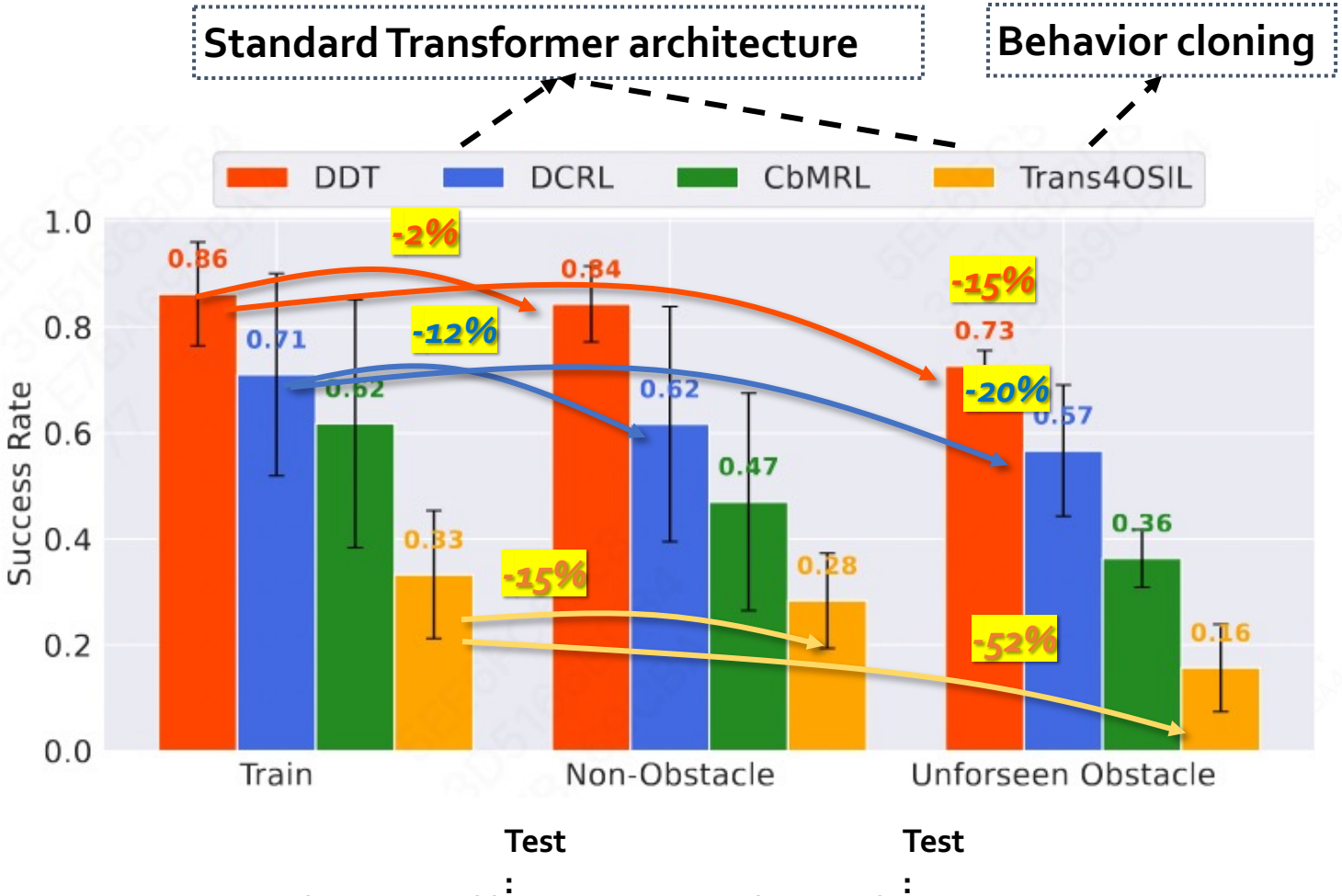


(B) Meta-World

(C) Complex Planning Tasks of Robot Manipulation

Illustration of Major Experiments in this paper. (A) Illustration of the VPAM, which is a new benchmark for OSIL with unforseen changes. The imitation points are provided by our DDT method. (B) Illustration of tasks in Meta-Wolrd. (C) Various Complex tasks of robot manipulation in clutter environments. (a): Grasp the blocked target object (cyan). (b): Stack the objects. (c): Collect the objects scattered over the desk together to the specified area (yellow).

RQ1: One-Shot Imitation Ability in Unseen Situations



Group results averged by 8 settings with 3 seeds (VPAM env).

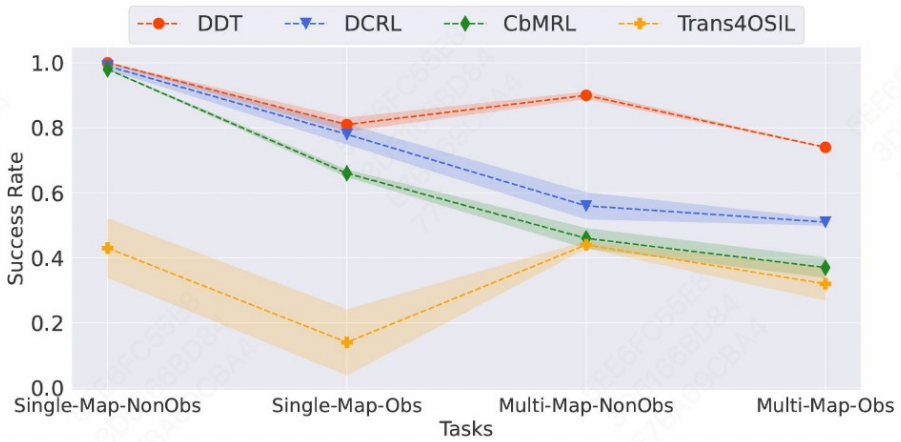
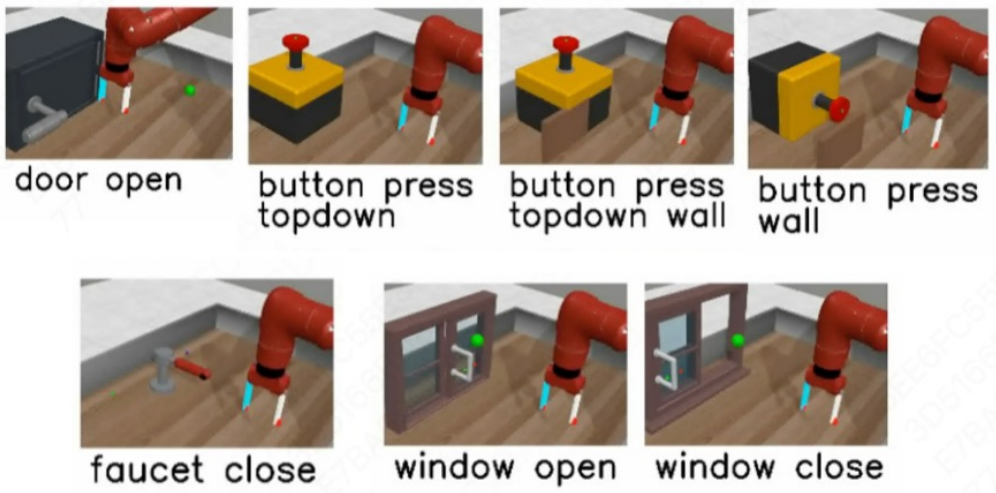


Illustration of the imitation policies' training performance among different settings. The colored areas denote the standard error among the three seeds. DDT displayed a **stable and better performance even in the training tasks**. We attribute this to the integration of the demonstration transformer architecture. This architecture conferred an additional training efficiency boost by implicitly introducing prior knowledge of how OSIL was achieved, facilitating easier adaptation across various tasks and settings with different complexities.

better generlization ability in unseen situations

consistent training performance among different tasks

RQ1: One-Shot Imitation Ability in Unseen Situations



Runtime Imitation

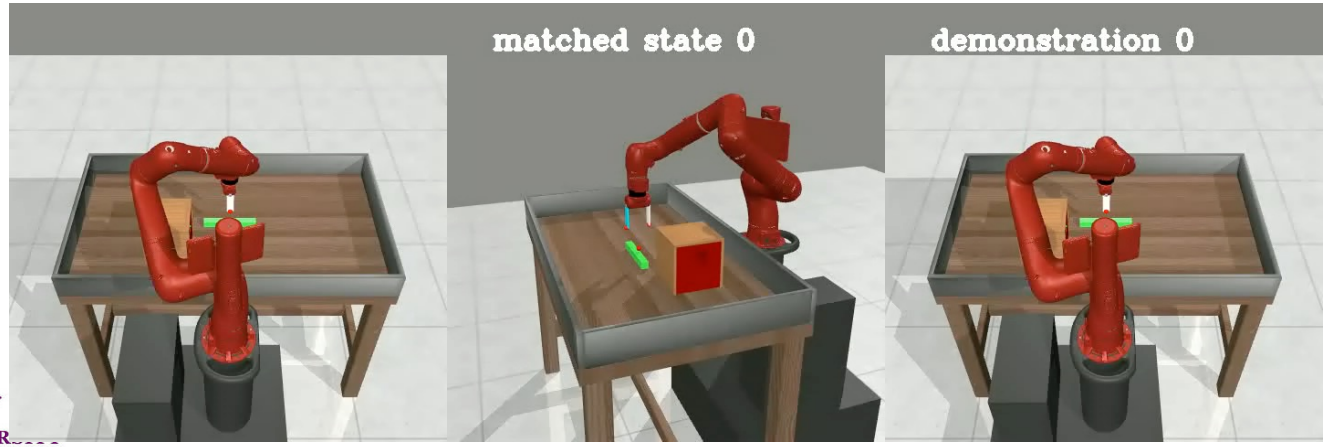
Table 4: Performance on unseen heterogeneous demonstrations.

Environment	Button Press	Door Close	Reach
Performance	0.78	1.00	0.75

Training tasks (reach 100% success rate)

Test tasks

We test and record the generalization performance on three types of unseen heterogeneous demonstrations with all positions of goals without fine-tuning.

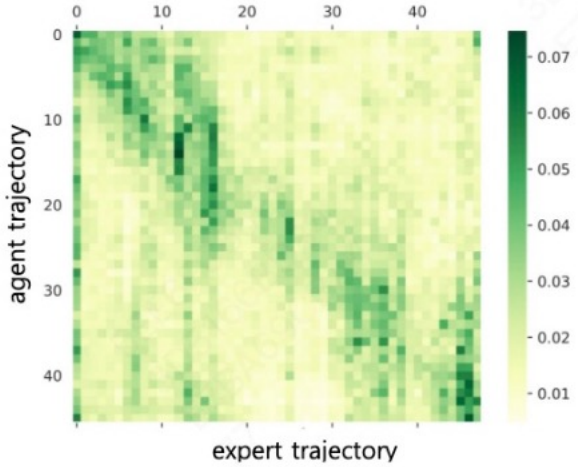


MORE VIDEO

SCAN ME

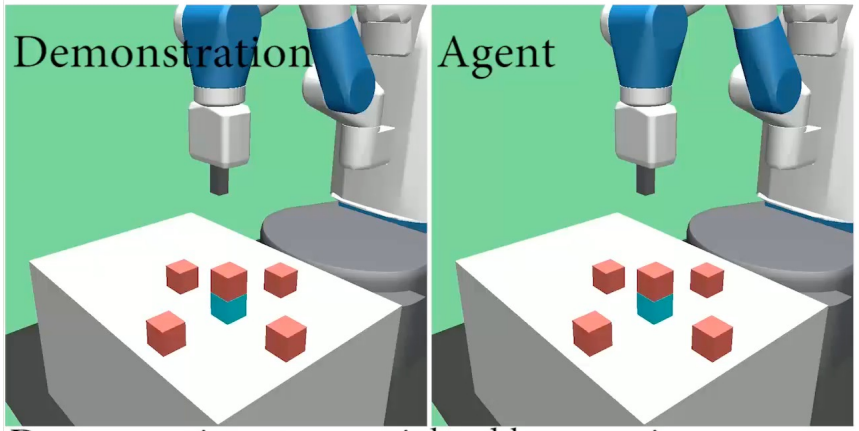
RQ2: Demonstration-Attention Mechanism for Demonstration Tracing in DDT

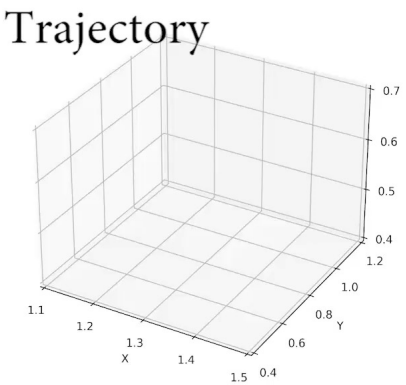
(a) Trajectory of DDT.



(b) Attention score.

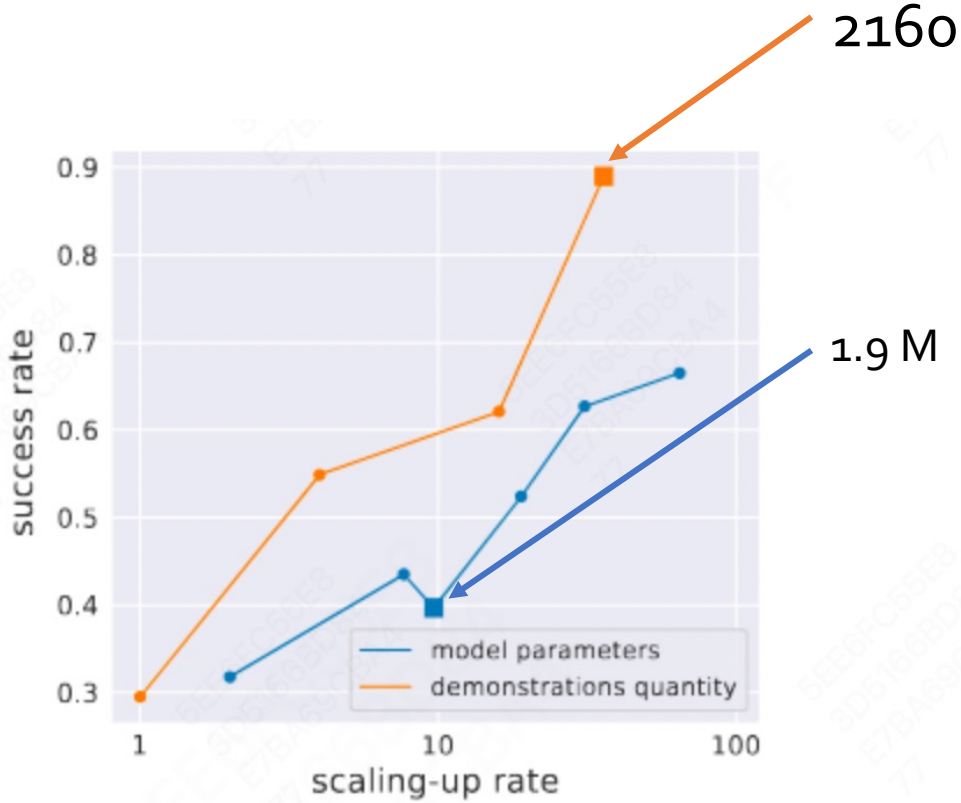
Trained on Multiple Tasks





Demonstration states weighted by attention scores:

RQ3: Similar Scaling Law of DDT when Scaling Up in the OSIL Setting



Asymptotic performance of DDT under varying demonstration quantities and model parameters, with each unit on the x-axis representing 60 demonstrations or 0.6 million parameters. The x-axis is on a logarithmic scale. **Square markers** depict the performance of the default DDT parameters.

Policy-conditioned Model: Generalizable Environment Model Learning

Policy-conditioned Environment Models are More Generalizable

Ruifeng Chen [*1,2] Xiong-Hui Chen [*1,2] Yihao Sun [1] Siyuan Xiao [1] Minhui Li [1] Yang Yu [1,2]

Policy-conditioned Model: Generalizable Environment Model Learning

1. **Background**
2. Motivation of our Solution
3. Experiment
4. Take-home Messages

Offline Environment Model Learning

Standard Offline Environment model Learning Objective

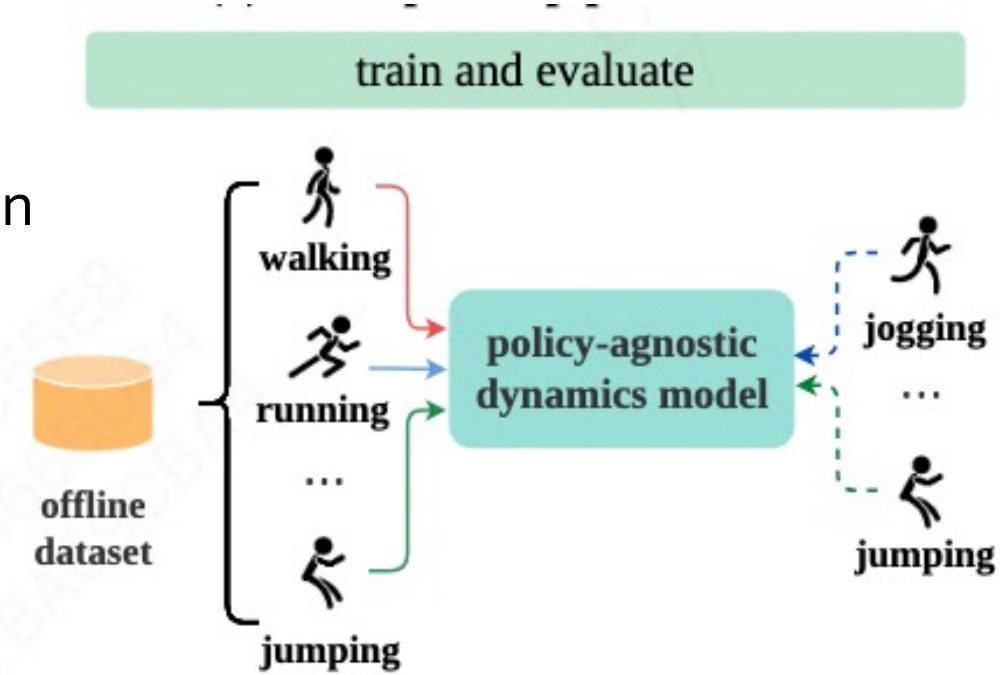
$$\hat{\mathrm{T}} = \arg\min_T \mathbb{E}_{s,a,s' \sim D}[-\log T(s'|s,a)]$$

The root purpose of environment model learning: Unseen Policy Evluation.

$$V_{\hat{T}}^{\pi} = \mathbb{E}_{s,a \sim \rho_{\hat{T}}^{\pi}}[r(s,a)]$$

$$\rho_{\hat{T}}^{\pi}(s,a,s') = \rho_{\hat{T}}^{\pi}(s)\pi(a|s)\hat{T}(s'|s,a)$$

$$\rho_{\hat{T}}^{\pi}(s) = (1-\gamma)\mathbb{E}_{s_0 \sim \rho_0}\left[\sum_{t=0}^{\infty}\gamma^t \Pr(s_t = s|s_0,\hat{T},\pi)\right]$$



Policy evaluation surpasses the SL's capability as it violates the independent and identically distributed i.i.d. assumption

Policy-conditioned Model: Generalizable Environment Model Learning

1. Background
1. **Solution**
2. Experiment
3. Take-home Messages

Motivation

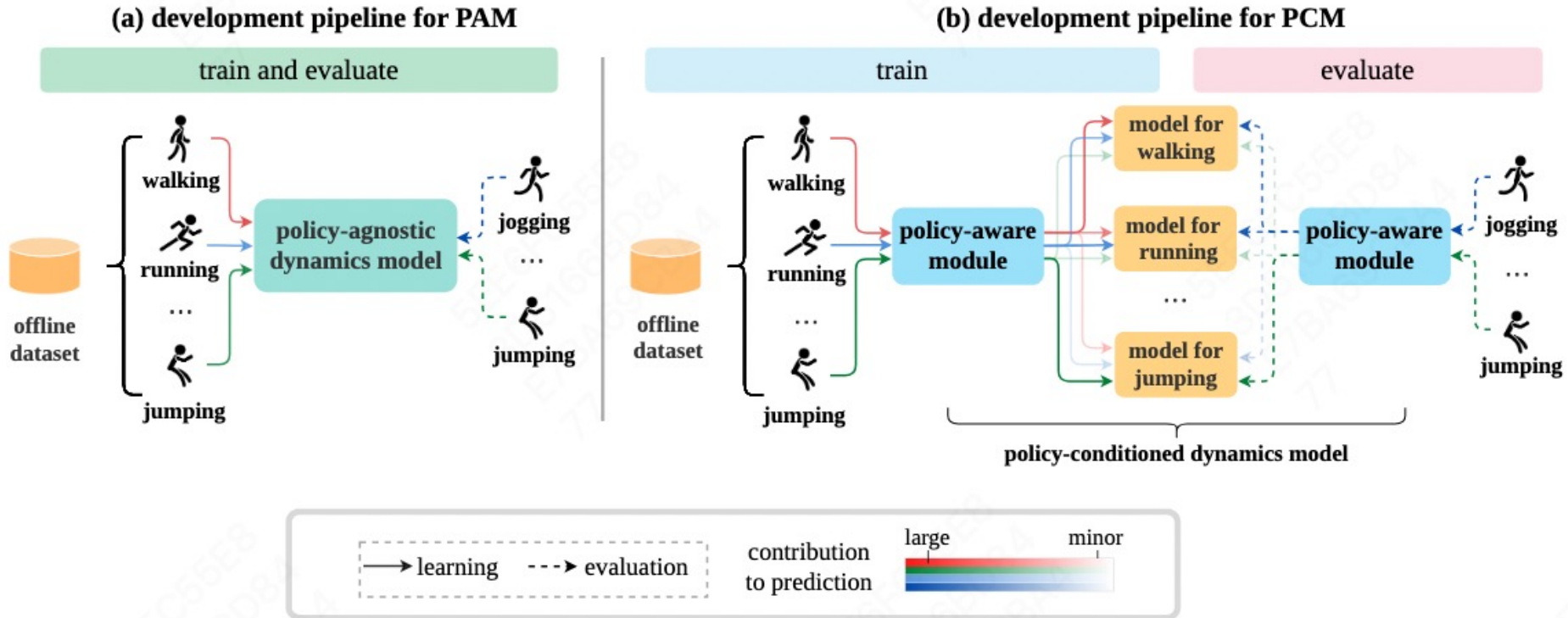


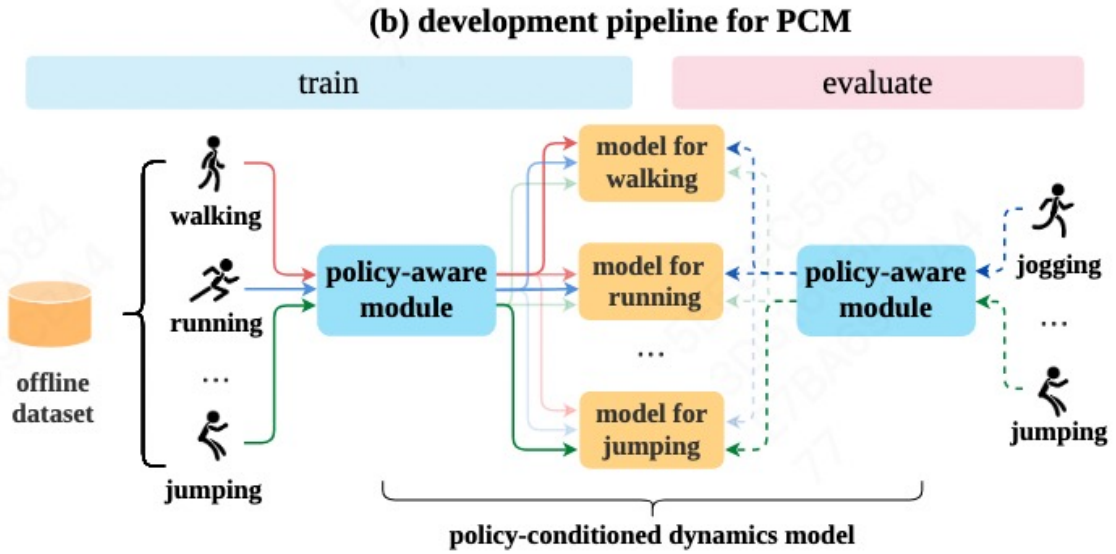
Figure 1: An illustration of the difference between the policy-agnostic model (left) and the policy-conditioned model (right). Suppose we wish to learn an environment where a biped robot is asked to move forward from an offline dataset including different locomotion patterns, such as jumping, walking, running, etc. Different locomotion patterns usually correspond to quite different transition patterns even though they can be regarded as a single task.

From the perpective of model usage, it is actually a multi-task problem.
-> Why can't we solved it by meta-learning?

Solution

Policy representation regularization

LSTM/transformer encoder



$$\min_{\phi,\theta,\psi} \mathbb{E}_{t\sim[0,H-2],\tau^{(j)}_{0:t+1}\sim\mathcal{D}}[-\log T_\psi(s_{t+1}|s_t,a_t,q_\phi(\tau^{(j)}_{0:t-1})) - \lambda\mathcal{R}_\pi(q_\phi(\tau^{(j)}_{0:t-1}),\pi^{(j)},\theta)], \quad (6)$$

where λ is a hyperparameter for policy-representation regularization. Since $\pi^{(j)}$ unknown in prior, in this work, we directly use the policy reconstruction term in implementation, i.e., $\mathcal{R}_\pi(q_\phi(\tau^{(j)}_{0:t-1}),\pi^{(j)},\theta) = \log p_\theta(a_t|s_t,q_\phi(\tau^{(j)}_{0:t-1}))$, where $p_\theta(a_t|s_t,z_t)$ is the jointed optimized policy decoder.

What is the benefits?

Theoretical analysis: The aaptation to policies reduces the PCMs' generalization error compared with standard PAMs.

$$l(\pi,T^*,T_{\hat{F}(\pi)}) \le \min_{\mu_i\in\Omega}\{\underbrace{l(\mu_i,T^*,T_{\hat{F}(\mu_i)})}_{\text{training error}} + \underbrace{L\cdot W_1(\rho^\pi,\rho^{\mu_i}) - C(\pi,\mu_i)}_{\text{generalization error}}\}$$

$$C(\pi,\mu_i) := l(\pi,T^*,T_{\hat{F}(\mu_i)}) - l(\pi,T^*,T_{\hat{F}(\pi)})$$

Policy-conditioned Model: Generalizable Environment Model Learning

1. Background

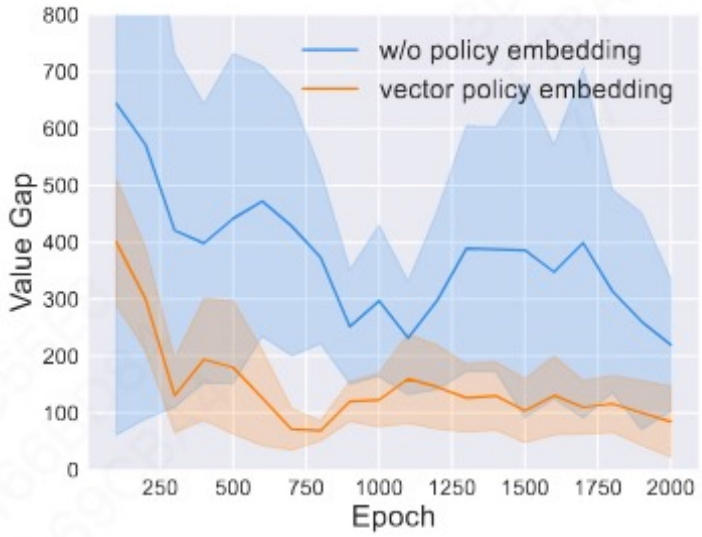
1. Solution

2. **Experiment**

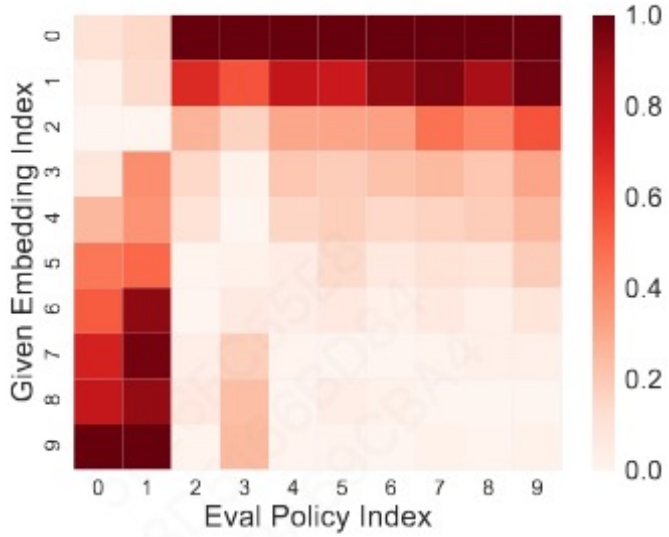
3. Take-home Messages

Expeirment: Proof-of-Concept Verification

Training error:
(vector rep.)

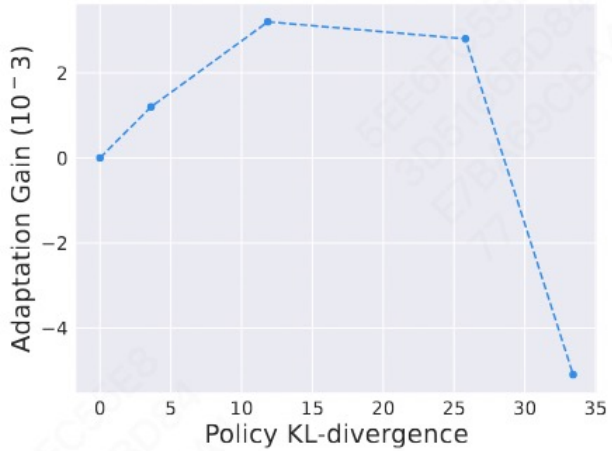


(b) Value gap

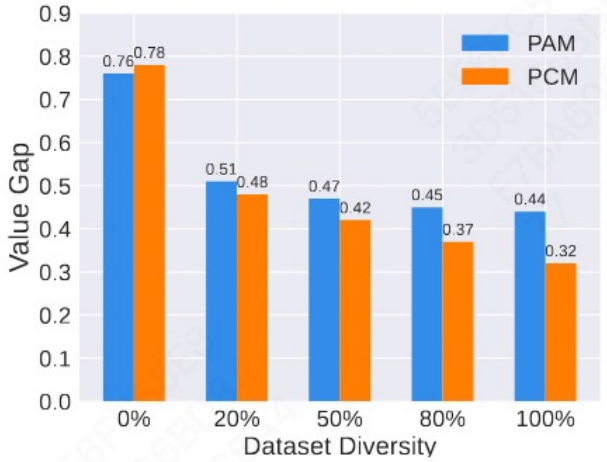


(c) Heatmap

Adaptation gain:
(lstm rep.)



(a) Adaptation gain

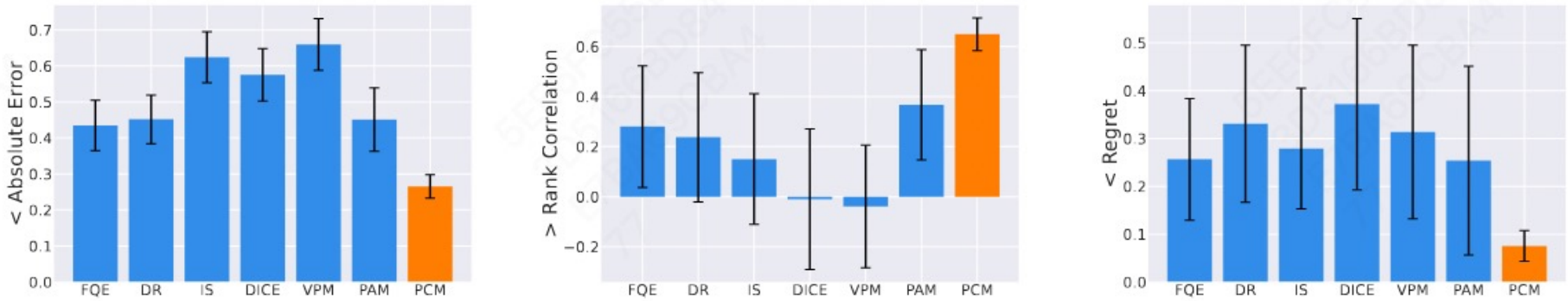


(b) Value gaps

Expeirment: Policy evaluation / selection

Table 1: Performance gain of offline policy selection for MOPO (Yu et al., 2020) by different methods.

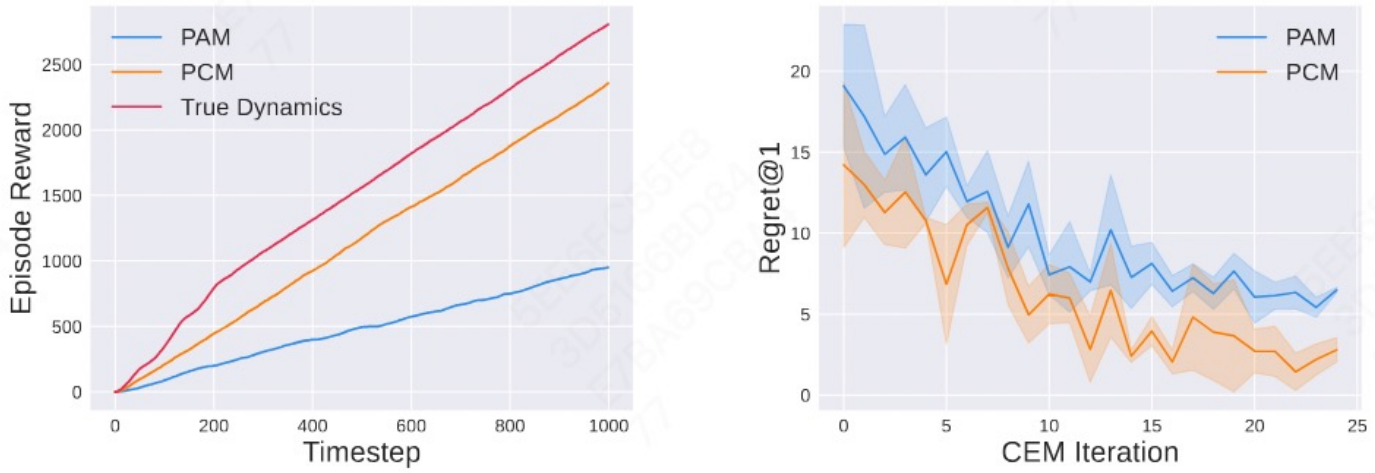
Task Name	Last Epoch	FQE	IS	DICE	PAM	PCM (Ours)
halfcheetah-medium-replay	39.3%	23.0%	87.8%	1.6%	1.6%	**98.4%**
hopper-medium-replay	56.0%	34.1%	56.0%	19.8%	47.3%	**64.8%**
walker2d-medium-replay	-4.6%	4.6%	34.3%	13.0%	-30.6%	**51.9%**
Average	30.2%	20.6%	59.4%	39.3%	11.5%	**71.7%**



(a) Absolute error (b) Rank correlation (c) Regret

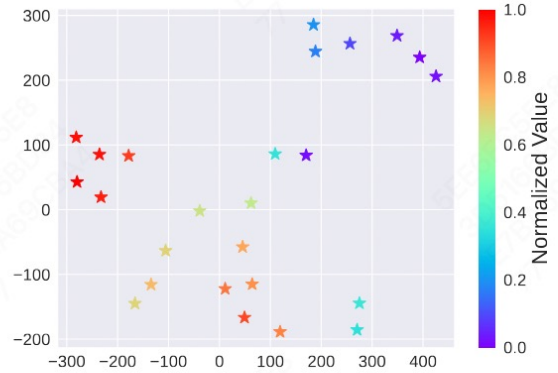
Figure 4: The performance of OPE in three metrics. To aggregate across tasks, we normalize the real policy values and evaluate policy values to range between 0 and 1. The error bars denote the standard errors among the tasks with three seeds.

Expeirment: MPC

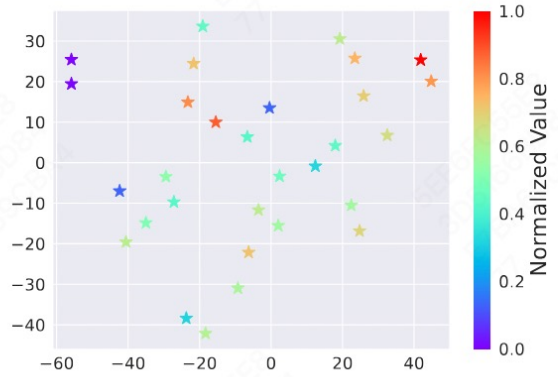


(a) Performance (b) Regret

Figure 5: Left shows cumulative rewards within an episode in HalfCheetah. Right shows regrets of PAM and PCM during CEM, obtained by tracking several planning processes.



(a) with representation loss



(b) without representation loss

Figure 6: Visualization for policy representations of different policies learned by PCM in HalfCheetah. Points are colored according to the normalized value.

Policy-conditioned Model: Generalizable Environment Model Learning

1. Background
2. Solution
3. Experiment
4. **Take-home Messages**

Take-home Messages

1. 万物皆可“META Learning by Context” [1,2,3]
2. 除了大力飞砖，提高泛化能力，我们能做的还有很多：
    1. Transformer远非完美，符合场景先验的模型架构 -> 更好的泛化能力；
    2. Supervised Learning远非完美-> 强化学习 with 扰动训练 -> 更好的泛化能力 [4]

[1] Offline Model-based Adaptable Policy Learning. 2021 NeurIPS
[2] Sim2Rec: A Simulator-based Decision-making Approach to Optimize Real-world Long-term User Engagement in Sequential Recommender Systems. 2023 ICDE
[3] Policy Rehearsing: Training Generalizable Policies for Reinforcement Learning. 2024 ICLR.
[4] Adversarial Counterfactual Environment Model Learning. 2023 NeurIPS

>> Thanks