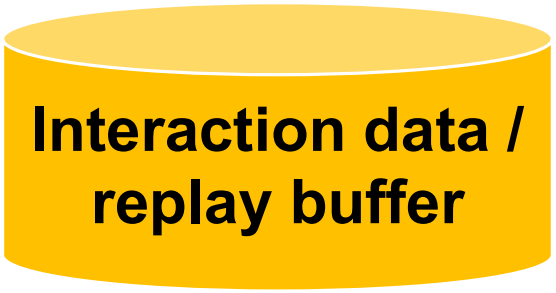
What Does Flow-Matching Bring to TD-Learning?

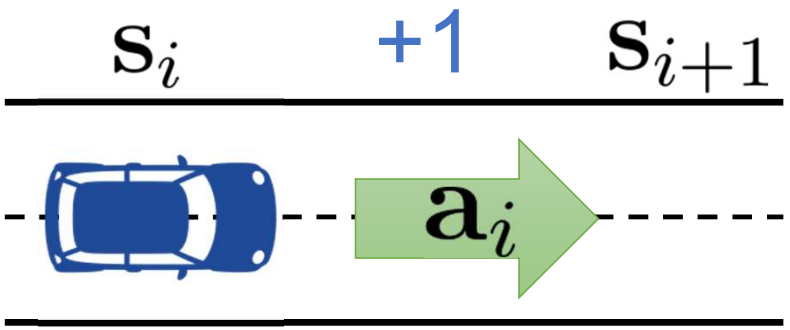
Bhavya Agrawalla, Michal Nauman, Aviral Kumar

Carnegie Mellon University
School of Computer Science

UNIVERSITY OF WARSAW

Primer: Notation and Terminology.





Action taken by **data collection** strategy

$$\{\mathbf{s}_i, \mathbf{a}_i, r(\mathbf{s}_i, \mathbf{a}_i), \mathbf{s}_{i+1}\}_{i=1}^{n}$$

observation (state of the world)

reward *value* (how good this action was)

next observation

Primer: TD-Learning.

Bellman equation

$$Q^{\pi}(\mathbf{s}_i, \mathbf{a}_i) := r(\mathbf{s}_i, \mathbf{a}_i) + \gamma Q^{\pi}(\mathbf{s}_{i+1}, \mathbf{a}_{i+1})$$

How can we learn to estimate Q?

Convert into a frozen target

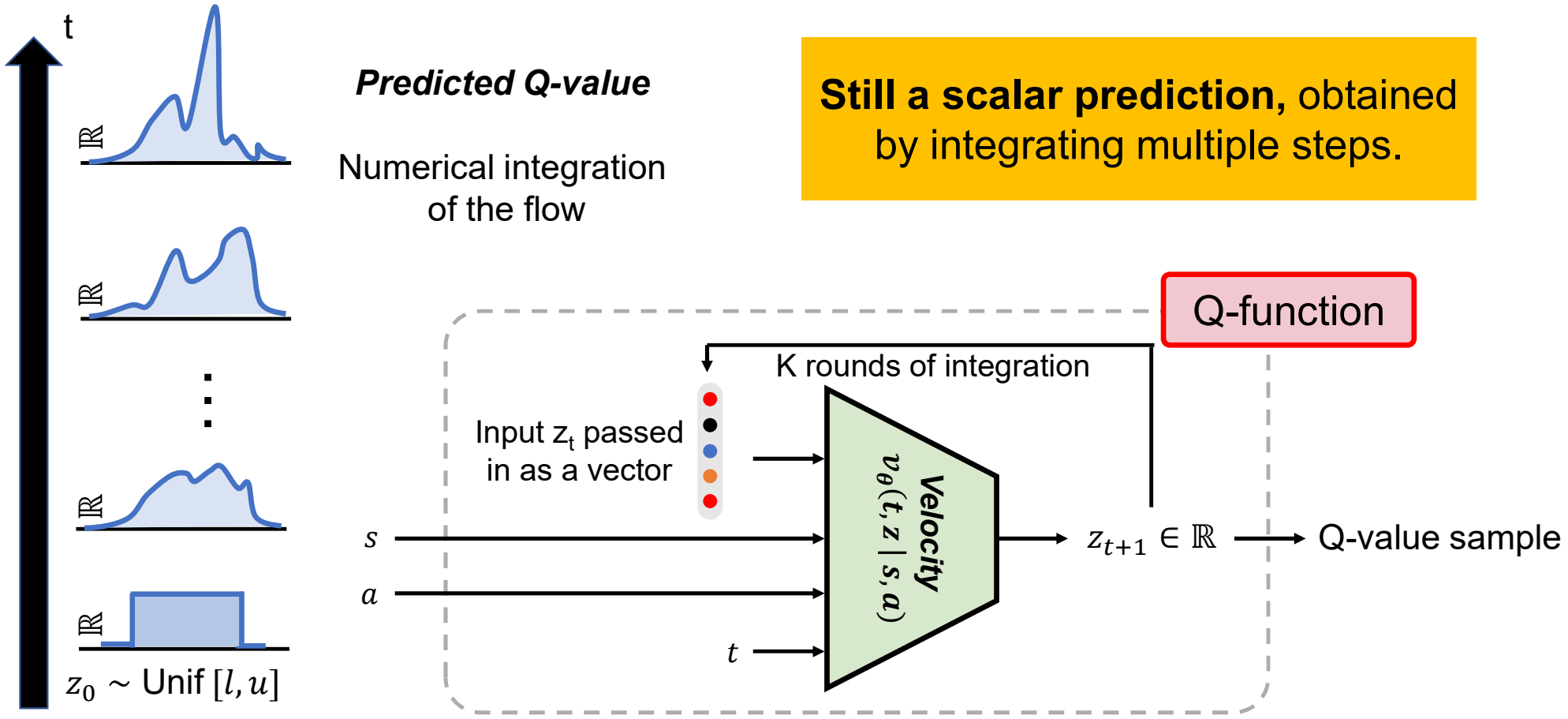
$$y(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}_i, \mathbf{a}_i) + \gamma Q_{\theta^{\text{old}}}(\mathbf{s}_{i+1}, \mathbf{a}_{i+1})$$

Training loss

$$\min_{\theta} \quad \mathbb{E}_{\text{data}} \left[\left(Q_{\theta}(\mathbf{s}_i, \mathbf{a}_i) - y(\mathbf{s}_i, \mathbf{a}_i) \right)^2 \right]$$

(and periodically update the target network)

Primer: Flow-Matching Q-Functions (`floq`).

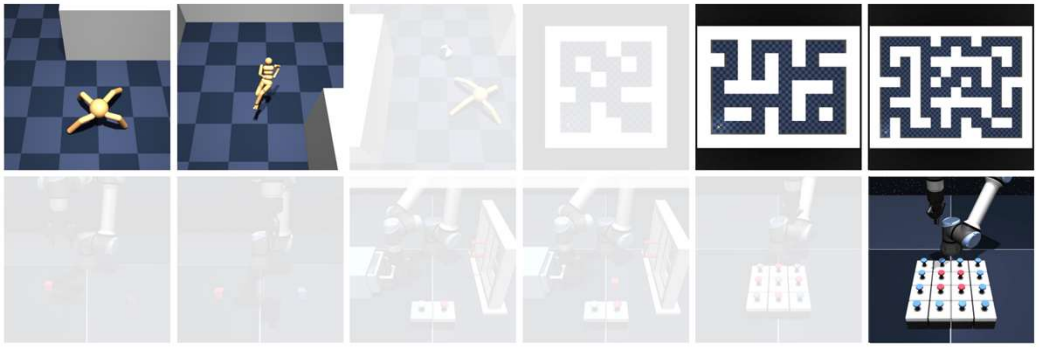


Agrawalla et al. **`floq`: Training Critics via Flow-Matching for Scaling Compute in Value-Based RL.** ICLR 2026.

floq Performance.

Simulated offline RL benchmarks

(***Hard tasks***: *precision is key*)



Environment (5 tasks each)	Gaussian Policy		Flow Policy			Flow Q-function (Ours)		
	BC	ReBRAC	SORL	FQL (1M)	FQL(2M)	floq (Def.)	floq (Best)	
antmaze-large	11 ±1	81 ±5	89 ±2	79 ±3	83 ±5	**91** ±5	**91** ±5	
antmaze-giant *(Hard)*	0 ±0	26 ±8	9 ±6	22 ±19	27 ±23	36 ±21	**51** ±12	**~2x**
hmmaze-medium	2 ±1	22 ±8	64 ±4	57 ±5	69 ±20	**82** ±10	**82** ±10	
hmmaze-large *(Hard)*	1 ±0	2 ±1	5 ±2	9 ±6	16 ±9	**28** ±9	**28** ±9	**~1.75x**
antsoccer-arena	1 ±0	0 ±0	**69** ±2	60 ±2	61 ±10	65 ±12	65 ±12	
cube-single	5 ±1	91 ±2	97 ±1	96 ±1	94 ±5	**98** ±3	**98** ±3	
cube-double *(Hard)*	2 ±1	12 ±1	25 ±3	29 ±2	25 ±6	**47** ±15	**47** ±15	**~2x**
scene	5 ±1	41 ±3	57 ±2	56 ±2	57 ±4	**58** ±6	**58** ±6	
puzzle-3x3 *(Hard)*	2 ±0	21 ±1	– ±–	30 ±1	29 ±5	**37** ±7	**37** ±7	
puzzle-4x4 *(Hard)*	0 ±0	14 ±1	– ±–	17 ±2	9 ±3	21 ±5	**28** ±6	**~3x**
Average Score (All Environments)	3	31	–	46	47	56	**59**	
Average Score (Hard Environments)	1	15	–	21	21	34	**38**	

Agrawalla et al. **floq: Training Critics via Flow-Matching for Scaling Compute in Value-Based RL.** ICLR 2026.

Why is Flow-Matching Effective for Q-Learning?

Recent works [1,2,3,..] show significant gains with flow-matching Q-functions.

Why are flow-matching Q-functions effective? Our findings:

1. **Distribution modelling** does **NOT** explain their success.

2. **Integration** enables **test-time recovery (TTR)** from imperfect value estimates.

3. **Flow-matching loss** enables **plasticity** under non-stationary TD targets.

[1] Agrawalla et. al, floq: Training Critics via Flow-Matching for scaling compute in Value-Based RL.
[2] Dong et. al, Value flows.
[3] Espinosa-Dice et. al, Expressive Value-Learning for scalable offline RL.

Distributional RL is an Insufficient Explanation.

1. **Expected-value TD backups (E) often outperform distributional RL backups (D).**
2. **Examples: Success rate is 52 v/s 30 (hmmaze-large), 86 v/s 74 (antmaze-giant).**
3. **Variance of final Q-values is much lower for E (last column), further supporting the claim.**

Table 1: ***Comparing expected-value (E) vs. distributional (D)*** flow-matching critics on representative OGBench tasks. Each entry reports **E** / **D**. While both variants learn similar expected Q-values, **D** produces higher-variance estimates (that more closely reflect variation in the return distribution) but does not outperform **E**.

Env.	Success (%)	$Q_\theta(\mathbf{s}, \mathbf{a})$	$\mathrm{Var}_{\mathbf{z}}(Q)$
hmmaze-large	**52** / 30	$-180/-170$	$0.2/\mathbf{4.5}$
antmaze-giant	**86** / 74	$-190/-200$	$0.1/\mathbf{0.7}$
cube-double	72 / 72	$-130/-130$	$1.1/\mathbf{6.3}$
hmmaze-medium	94 / 94	$-170/-170$	$0.3/\mathbf{2.3}$

Agrawalla et al. **What Does Flow-Matching Bring to TD-Learning?** arXiv 2026.

floq Learns to "Recover" from Noisy Q-Values

1. **When training against moving TD-targets, flow-critics learn to use the integration to correct errors. ("test-time recovery").**

2. **As a result, they are more robust to noisy targets and learn plastic features.**

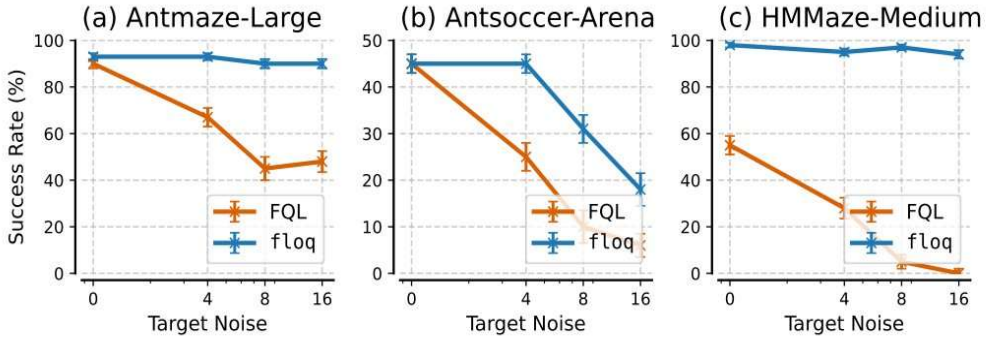
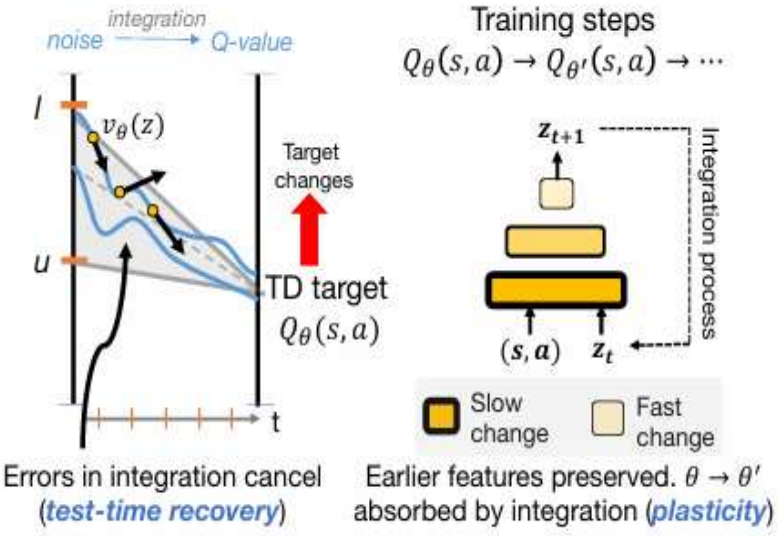
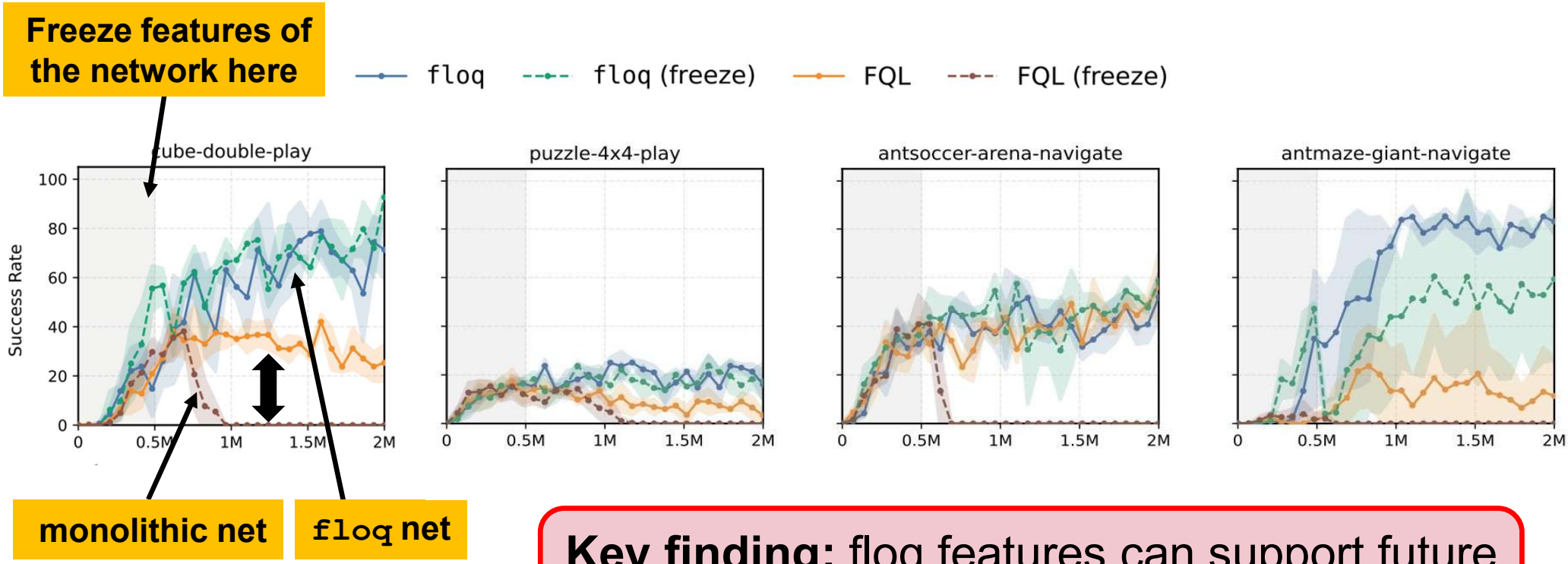


Figure 2: ***Performance of flow-matching (floq) and monolithic (FQL) critics when trained with target noise.*** Observe that flow-matching critics are much more robust to noise in TD targets, while performance of FQL (monolithic critic) degrades substantially faster, even when they start at a similar point (antmaze/antsoccer).



Flow-Matching Enables Plastic Feature Learning.



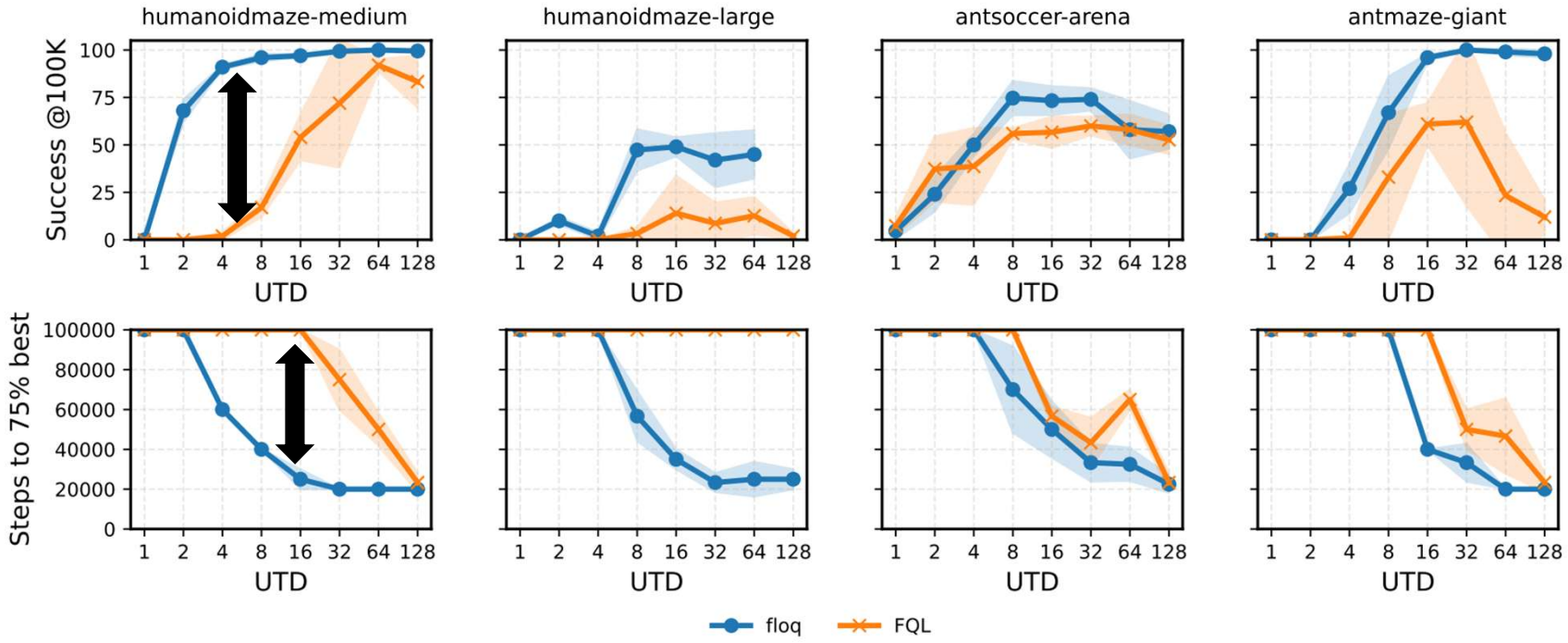
Key finding: floq features can support future targets (i.e., "***plastic***" features)

Agrawalla et al. **What Does Flow-Matching Bring to TD Learning?** arXiv 2026.

Consequences of Plastic Feature Learning.

Success w/ limited data

Sample efficiency



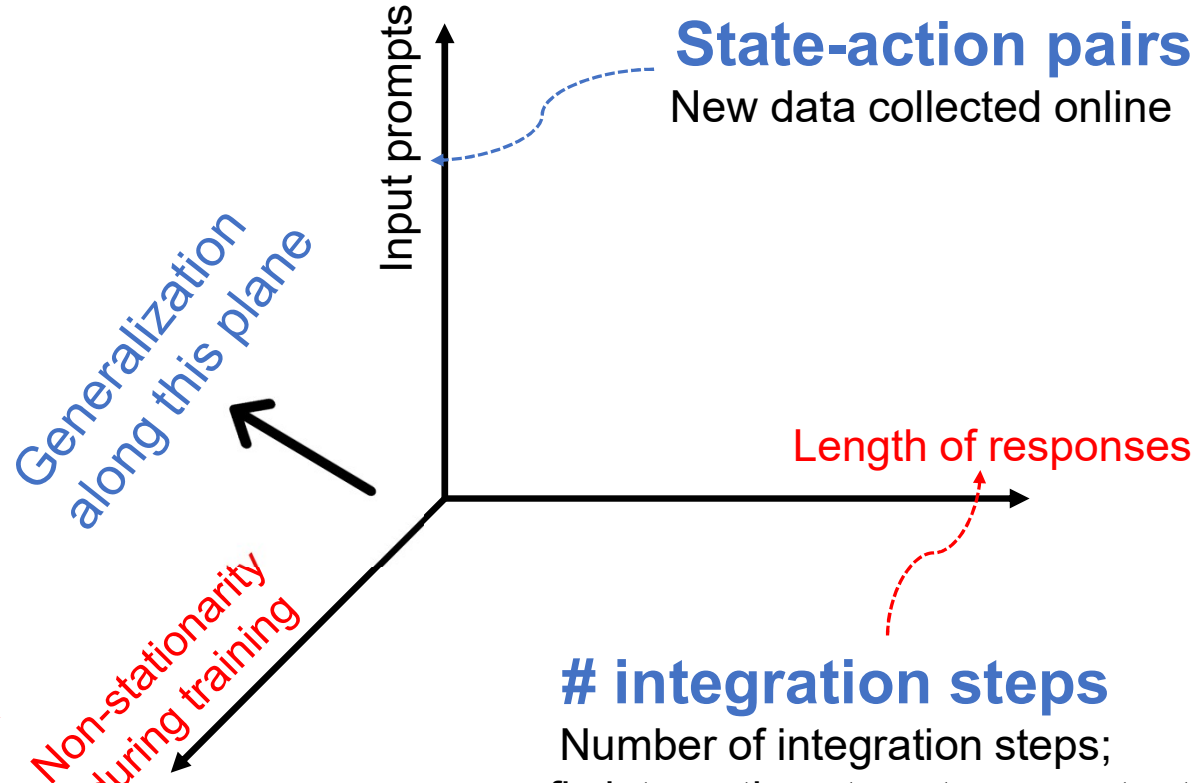
Key finding: Can improve with online experience in a data constrained regime ***very fast***, with aggressive updates!

Agrawalla et al. **What Does Flow-Matching Bring to TD Learning?** arXiv 2026.

Axes of Generalization in `floq` Networks

For `floq` nets

- New (s, a) pairs (yes!)
- ~~Longer lengths (no!)~~
- Training time (yes!)



State-action pairs
New data collected online

Non-stationarity
Moving targets and new data collected online in the environment

 # integration steps
Number of integration steps; we fix integration steps to a constant and do supervised training

Agrawalla et al. **What Does Flow-Matching Bring to TD Learning?** arXiv 2026.

Summary & Takeaways

- Training with a flow-matching loss presents a useful way to utilize **iterative test-time compute** for modeling non-stationary objects.

- Connections to test-time compute and ***plastic features*** from LLM land.

- **But still some open questions**:
  - What does exploration in a CoT correspond to for flows?
  - RL vs SFT gap in LLMs – what does it mean here?

Thank You!