Memory Augmented Policy Optimization (MAPO) for Program Synthesis and Semantic Parsing

Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc Le, Ni Lao

NORTHWESTERN UNIVERSITY
AI2 mosaix

Program Synthesis / Semantic Parsing

Problem: Learning to generate sequences $\vec{a}$ with high return $R(\vec{a})$ via policy optimization

($\vec{a}$ = a program)

($R(\vec{a})$ = correct or not)

$$O(\pi) = \sum_{\vec{a} \in \mathcal{A}} \pi(\vec{a}) R(\vec{a})$$

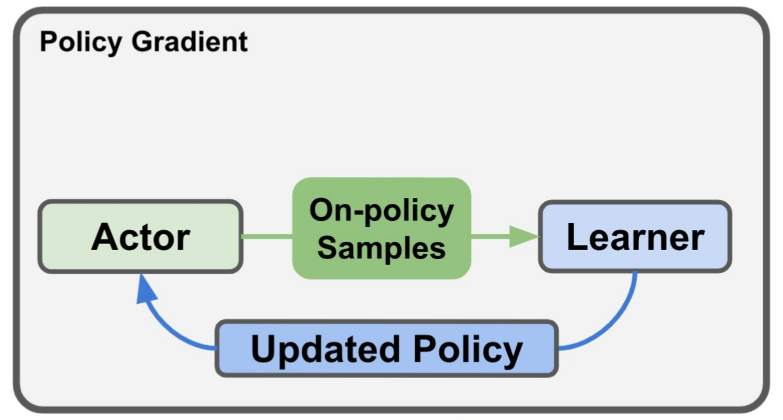
how many more passengers flew to los angeles than to saskatoon?

Rank	City	Passengers	Ranking	Airline
1	United States, Los Angeles	14,749		Alaska Airlines
2	United States, Houston	5,465		United Express
3	Canada, Calgary	3,761		Air Transat, WestJet
4	Canada, Saskatoon	2,282	4	
5	Canada, Vancouver	2,103		Air Transat
6	United States, Phoenix	1,829	1	US Airways
7	Canada, Toronto	1,202	1	Air Transat, CanJet
8	Canada, Edmonton	110		
9	United States, Oakland	107		

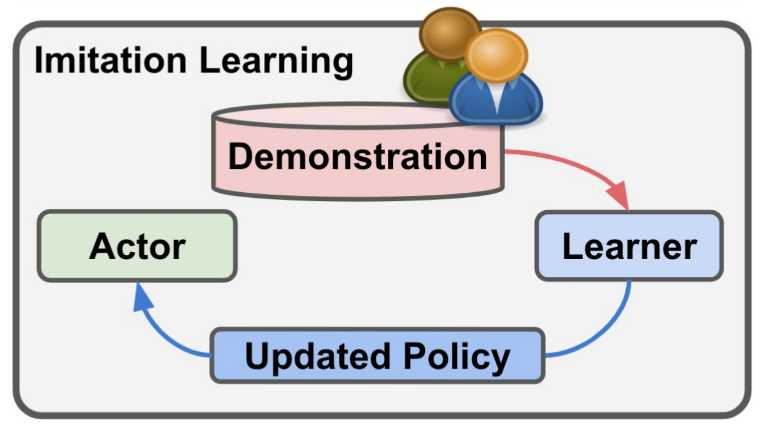

```
(filter_in rows ['saskatoon'] r.city)
(filter_in rows ['los angeles'] r.city)
(diff v1 v0 r.passengers)
```

Latent

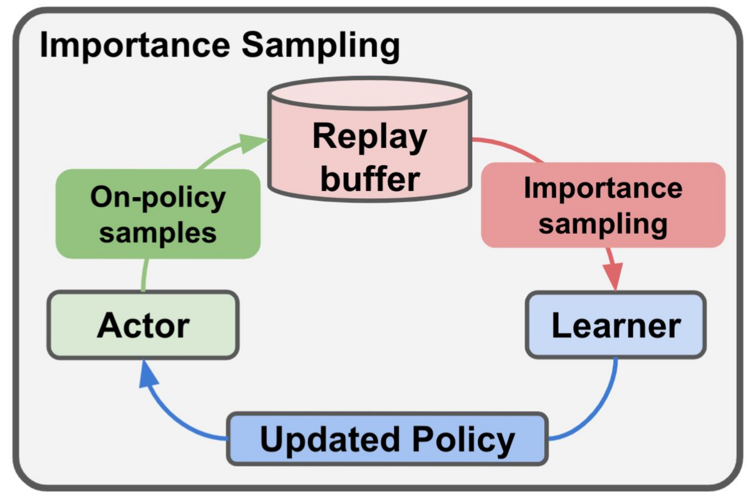
Existing Solutions



- ✅ Unbiased => optimal solution
- ❌ High variance => slow training
- ✅ Only requires a reward signal



- ❌ Biased => suboptimal solution
- ✅ Low variance => fast training
- ❌ Requires human supervision



With truncation	W/o truncation
❌ Biased	✅ Unbiased
✅ Low variance	❌ High variance

- ✅ Only requires a reward signal

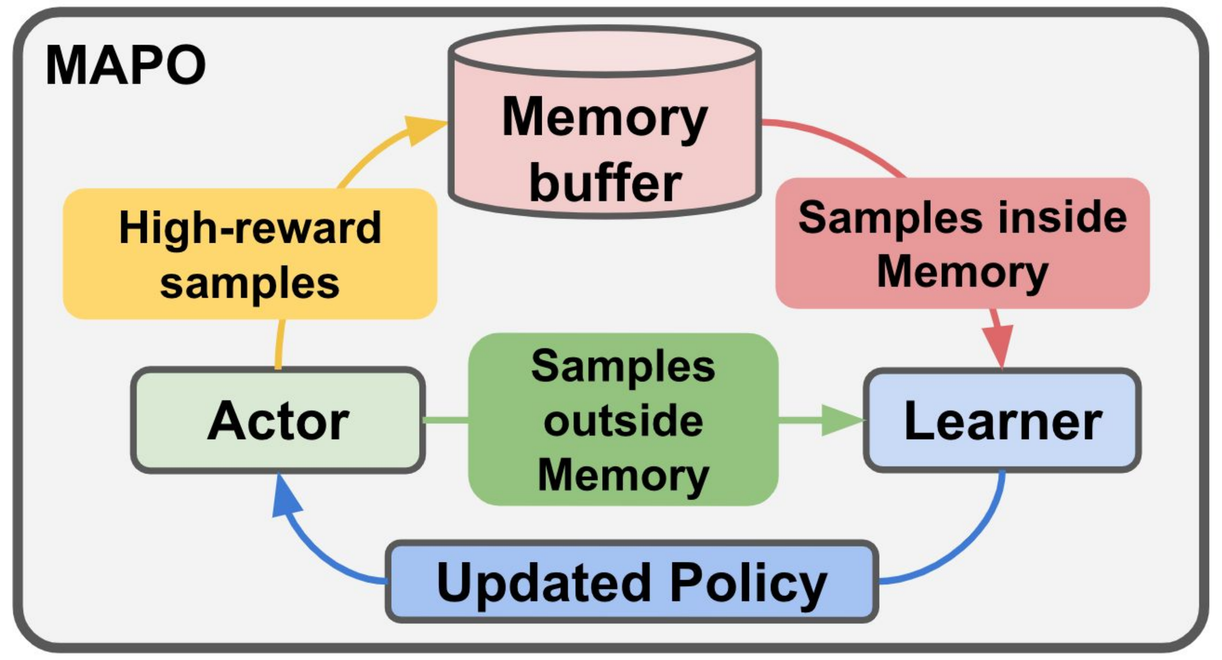
Memory Augmented Policy Optimization

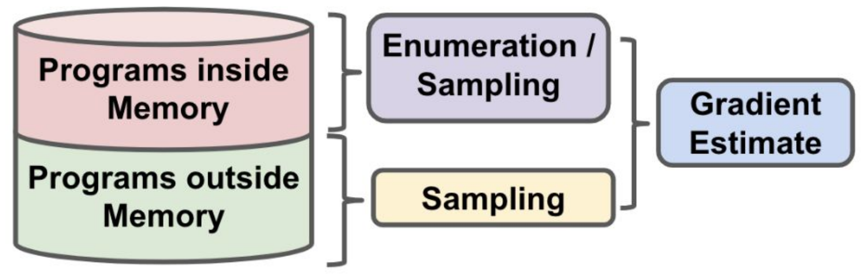
Key idea: Express the expected return objective as the sum of two expectations inside and outside a memory buffer of sequences $\mathcal{B} \equiv \{(\vec{a}_i, r_i)\}_{i=1}^{N}$

$$O(\pi) = \underbrace{\sum_{\vec{a}_i \in \mathcal{B}} \pi(\vec{a}_i) r_i}_{\text{Expectation inside } \mathcal{B}} + \underbrace{\sum_{\vec{a} \notin \mathcal{B}} \pi(\vec{a}) R(\vec{a})}_{\text{Expectation outside } \mathcal{B}}$$

- **MAPO** incorporates a memory buffer of promising sequences to compute an unbiased gradient estimate with low variance.

$$\nabla_\theta O(\pi_\theta) = \underbrace{\sum_{\vec{a}_i \in \mathcal{B}} \pi_\theta(\vec{a}_i) r_i \nabla_\theta \log \pi_\theta(\vec{a}_i)}_{\text{Expectation inside } \mathcal{B}} + \underbrace{\sum_{\vec{a} \notin \mathcal{B}} \pi_\theta(\vec{a}) R(\vec{a}) \nabla_\theta \log \pi_\theta(\vec{a})}_{\text{Expectation outside } \mathcal{B}}$$

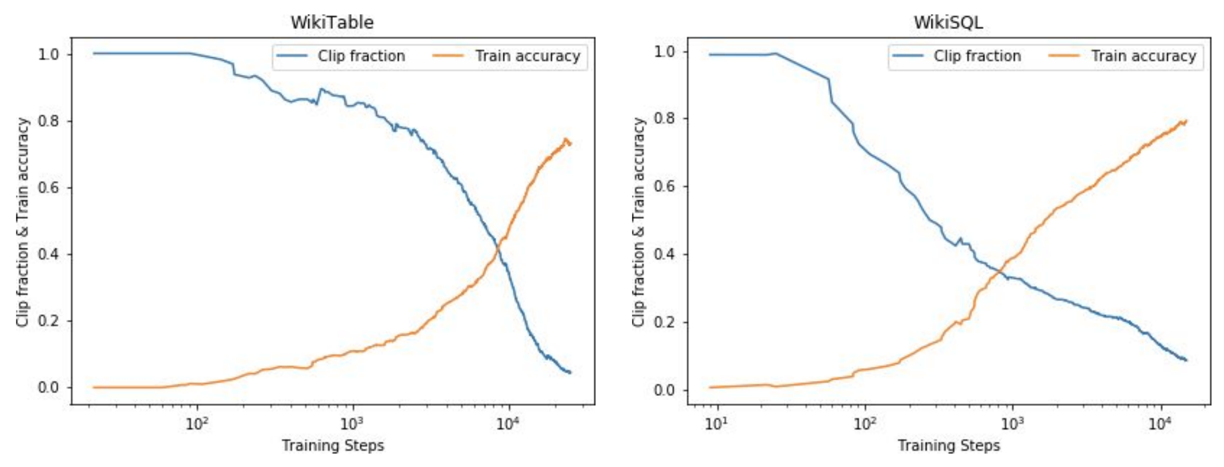




- ✅ Unbiased => optimal solution
- ✅ Low variance => fast training
- ✅ Only requires a reward signal

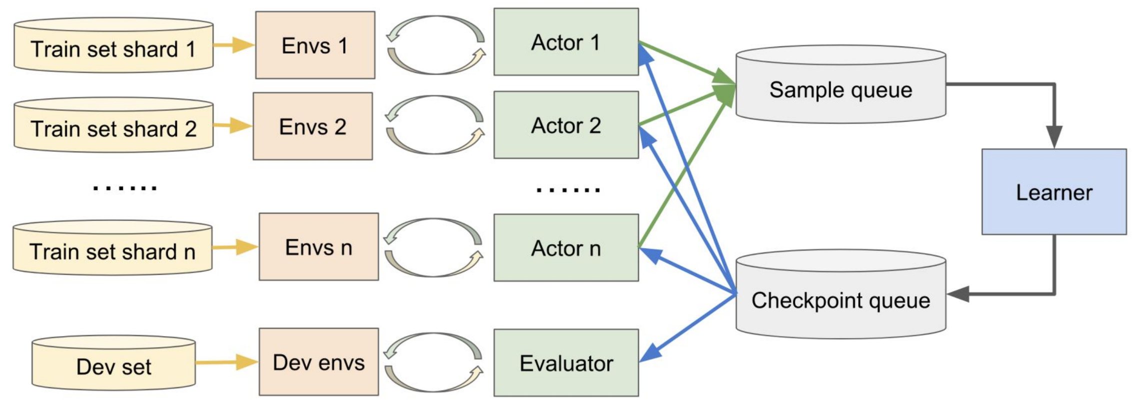
- **Memory weight clipping**
  - Force the training to pay attention to the memory by clipping the weight.
  - Trade off bias in the initial stage for faster training.

$$\pi^c_{\mathcal{B}} = \max(\pi_{\mathcal{B}}, \alpha)$$



- **Systematic exploration**
  - Use a bloom filter to force the exploration to generate new programs.
  - Trade off memory for more efficient exploration.

- **Distributed sampling**
  - Distribute the cost of computing $\pi_{\mathcal{B}}$ and sampling into the actors.
  - Multiple actors each interacting with a shard of training set and send samples to a learner to update the model.

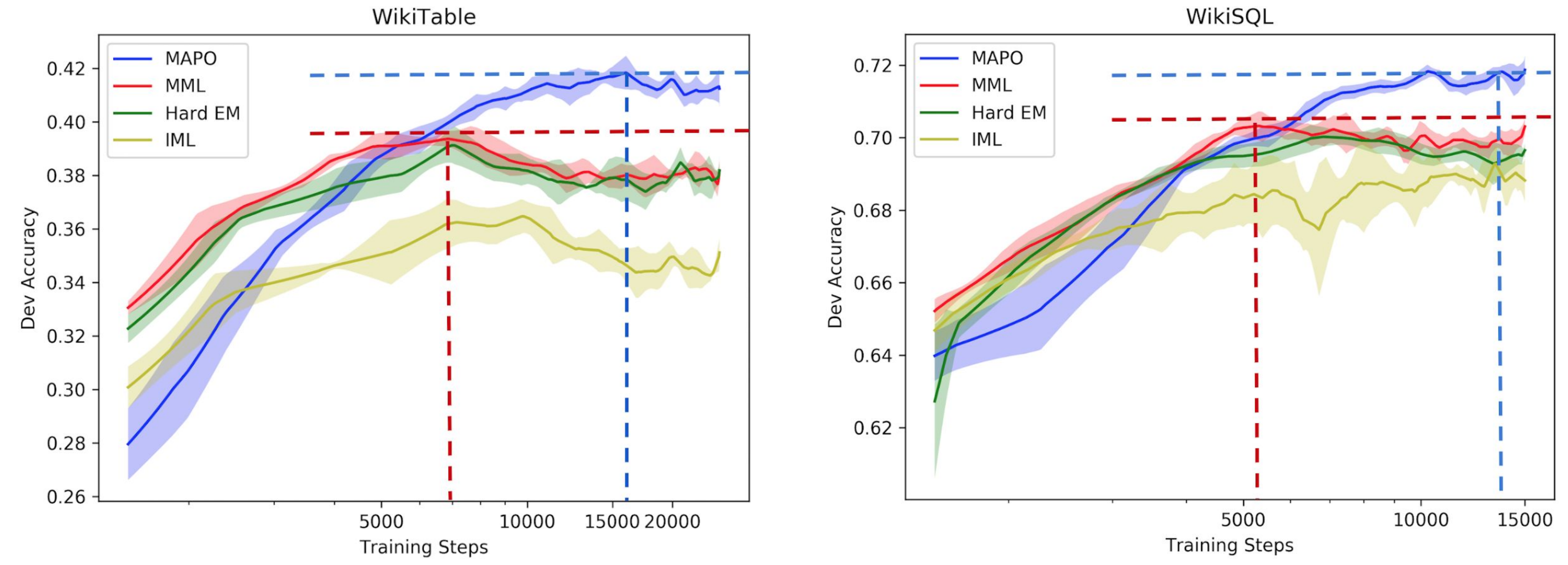


Experiments

	E.S.	Dev.	Test
Pasupat & Liang (2015)	-	37.0	37.1
Neelakantan *et al.* (2017)	1	34.1	34.2
Neelakantan *et al.* (2017)	15	37.5	37.7
Haug *et al.* (2017)	1	-	34.8
Haug *et al.* (2017)	15	-	38.7
Zhang *et al.* (2017)	-	40.4	43.7
MAPO	1	**42.4** ± 0.5	**43.2** ± 0.5
MAPO (ensembled)	10	-	**46.6**

Fully supervised		Dev.	Test
Zhong *et al.* (2017)	Strong supervision	60.8	59.4
Wang *et al.* (2017)		67.1	66.8
Xu *et al.* (2017)		69.8	68.0
Huang *et al.* (2018)		68.3	68.0
Yu *et al.* (2018)		74.5	73.5
Sun *et al.* (2018)		75.1	74.6
Dong & Lapata (2018)		**79.0**	**78.5**
Weakly supervised		Dev.	Test
MAPO		**71.6** ± 0.6	**71.8** ± 0.4
MAPO (ensemble of 5)		-	**74.9**

- First RL-based state-of-the-art method on ***WikiTableQuestions***.
- Competitive to state-of-the-art methods on ***WikiSQL***, which use strong supervision (the ground truth programs), while MAPO only uses weak supervision (the final answers).



- **MAPO** converges slower than **maximum likelihood training**, but reaches a better solution.
- **REINFORCE** doesn't make much progress (<10% accuracy).
- **Spurious programs**: right answer for the wrong reason

Rank	Nation	Gold	Silver	Bronze	Total
1	Nigeria	14	12	9	35
2	Algeria	9	4	4	17
3	Kenya	8	11	4	23
4	Ethiopia	2	4	7	13
5	Ghana	2	2	2	6
6	Ivory Coast	2	1	3	6
7	Egypt	2	1	0	3
8	Senegal	1	1	5	7

Which nation won the most silver medal?

- **Correct program**:

(argmax rows “Silver”)
(hop v1 “Nation”)

- **Spurious programs**:

(argmax rows “Gold”)
(hop v1 “Nation”)

(argmax rows “Bronze”)
(hop v1 “Nation”)

- **Comparison of MAPO, MML, IML with a simplified example**

	Question 1		Question 2	
	correct	spurious	spurious	spurious
Iterative Maximum Likelihood (IML)	0.5	0.5	0.5	0.5
Maximum Marginal Likelihood (MML)	0.8	0.2	0.5	0.5
MAPO	0.6	0.15	0.1	0.1
Model Probability	0.6	0.15	0.1	0.1