Guided Flow Policy

April 23, 2026

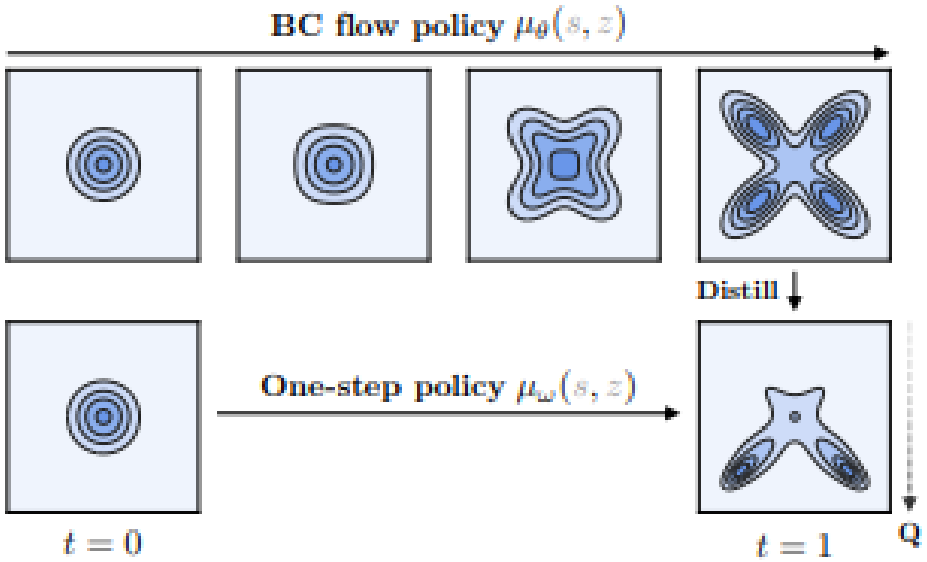
1. 오프라인 강화학습은 고정 데이터를 이용하여 안전하게 정책을 학습하나 이는 분포밖 데이터를 과대평가하는 문제가 생김
2. 이를 해결하기 위해 flow model을 이용하여 학습 데이터를 정확히 표현하고 이를 distillation하며 학습을 하는 연구가 진행 되어왔음.
3. 위 논문은 더 나아가 VaBC라는 정책을 통해 distillation 한 정책과 flow model의 정책이 서로 상호작용하며 높은 Q함수를 가지는 정책을 학습함

BRAC 알고리즘은 기존 actor-critic 알고리즘을 수정하여 학습하려는 고정데이터와 현재 학습중인 정책이 서로 멀어지지 않게 규제하는 항을 추가함

1. $L^C(\phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D},\ a'\sim\pi_\theta(\cdot|s')}\left[\left(Q_\phi(s,a) - r - \gamma Q_{\bar{\phi}}(s',a')\right)^2\right].$

   여기서 $\bar{\phi}$는 EMA로 천천히 학습되는 파라미터

2. $L^A(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D},\ a_\theta\sim\pi_\theta(\cdot|s)}\left[-Q_\phi(s,a_\theta) + \overbrace{\alpha\|a_\theta - a\|_2^2}^{\text{BC term}}\right]$



flow matching은 간단한 분포를 복잡한 분포로 이동시키는 모델로서, 기존의 FQL 논문에서는 이를 이용해 $z \sim N(0, I)$를 행동분포로 이동하는 모델을 학습하고, 그 모델을 distillation 해주는 손실함수를 BC term에 넣어 빠르게 정책을 학습하는 방법을 제시함

- $L_{\text{Flow}}(\theta) = \mathbb{E}_{\substack{s,\ a=x^1\sim\mathcal{D},\\ x^0\sim\mathcal{N}(0,I_d),\\ t\sim\text{Unif}([0,1])}} \left[\left\| v_\theta(t,s,x^t) - (x^1 - x^0) \right\|_2^2 \right].$

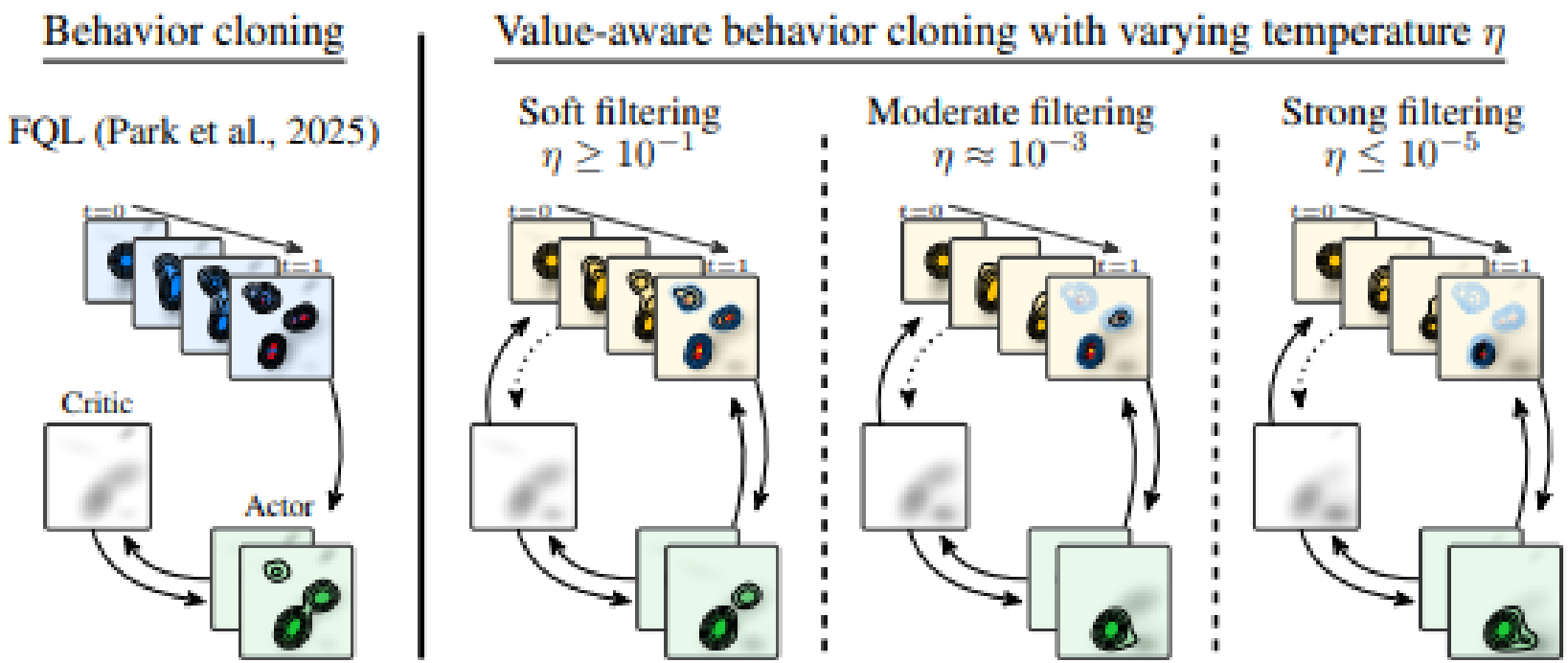
  v_θ 를통해 $z \sim N(0, I)$ 를 고정데이터의 액션으로 이동시키는 벡터장 학습

- $L_{\text{Distill}}(\omega) = \mathbb{E}_{\substack{s\sim\mathcal{D},\\ z\sim\mathcal{N}(0,I_d)}} \left[\left\| \mu_\omega(s,z) - \mu_\theta(s,z) \right\|_2^2 \right].$

  μ_θ 는 위에서 학습한 v_θ 를 적분해서 얻은 함수로, FLOW 정책에 대응되는 액션이 나옴. 또한 μ_ω를 통해 이를 distillation해주는 손실함수를 정의함

- $L_\pi(\omega) = \underbrace{\mathbb{E}_{s\sim\mathcal{D},\ a^\pi\sim\pi_\omega}\left[-Q_\phi(s,a^\pi)\right]}_{\text{Q loss}} + \underbrace{\alpha L_{\text{Distill}}(\omega)}_{\text{“BC” loss}}.$

  BRAC 알고리즘의 BCterm 에 위의 손실함수를 넣어서 학습함으로서, 적분하지 않아도 되는 정책을 학습한다.



위 논문에서는 Value-aware Behavior Cloning(VaBC) flow actor를 정의하여, distillaion actor와 flow actor가 서로 연결되게 하여 더 높은 Q-value를 가지는 액션을 학습한다

critic 함수 Q_ϕ, one step 정책 π_ω, VaBCpoliy π_θ 를 이용하여 학습

- critic 함수 Q_ϕ 학습

$$L^C(\phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D},\ a'\sim\pi_\theta(\cdot|s')}\left[\left(Q_\phi(s,a) - \overbrace{\left(r+\gamma Q_{\bar{\phi}}(s',a')\right)}^{\text{bellman target y}}\right)^2\right].$$

  이 때 좀 더 보수적인 값을 가지는 VaBC 이용하기도 한다.
  $y^{\text{VaBC}}(s,r,s') = r + \frac{\gamma}{2}\left(Q_{\bar{\phi}}(s',\mu_\theta(s',z)) + Q_{\bar{\phi}}(s',\mu_\omega(s',z))\right)(z\sim\mathcal{N}(0,I_d).$
- actor 함수학습을 통해 one step 정책 학습
  $L^A(\theta) = \mathbb{E}_{s\sim\mathcal{D},\ z\sim\mathcal{N}(0,I_d)}\left[-\lambda Q_\phi\left(s,\mu_\omega(s,z)\right) + \alpha\left\|\mu_\omega(s,z) - \mu_\theta(s,z)\right\|_2^2\right].)$
  기존에 학습시켜준 VaBC를 distillation 하는 손실함수를 BC term으로 두어 π_ω 를 학습한다

- VaBC policy를 학습한다

$$L^{\mathrm{VaBC}}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D},\ \epsilon\sim\mathcal{N}(0,I_d),\ t\sim\mathcal{U}([0,1])}\left[g_\eta(s,a)\,\|v_\theta(t,s,a_t) - (a-\epsilon)\|_2^2\right].$$

$$g_\eta(s,a) := \frac{\exp\left(\frac{\lambda}{\eta}Q_\phi(s,a)\right)}{\exp\left(\frac{\lambda}{\eta}Q_\phi(s,a)\right)+\exp\left(\frac{\lambda}{\eta}Q_\phi(s,\mu_\omega(s,z))\right)}, \qquad z\sim\mathcal{N}(0,I_d).$$

1. g_η 함수를 통하여 데이터에 있는 행동의 Q 값과, onestep 정책의 행동의 Q 값을 비교하여, 데이터에 있는 행동값이 큰 쪽에 가중치를 둔다.
2. 만약 행동값이 크다면 $g_\eta > 0.5$ 가 되어 영향력이 커지고, 작다면 $g_\eta < 0.5$ 면 영향력이 작아진다.
3. η 값이 작으면 g_η값에 더욱 민감해져 0 혹은 1처럼 binary 와 비슷해지고,
   η 값이 클수록 g_η값이 다양해진다.

Algorithm 1 Flow Q-Learning (FQL)

function $\mu_\theta(s, z)$ ▷ BC flow policy
  for $t = 0, 1, \ldots, M-1$ **do**
    $z \leftarrow z + v_\theta(t/M, s, z)/M$ ▷ Euler method
  return z

while not converged **do**
  Sample batch $\{(s, a, r, s')\} \sim \mathcal{D}$
  ▷ Train critic Q_ϕ
  $z \sim \mathcal{N}(0, I_d)$
  $a' \leftarrow \mu_\omega(s', z)$
  Update ϕ to minimize $\mathbb{E}[(Q_\phi(s, a) - r - \gamma Q_{\bar{\phi}}(s', a'))^2]$
  ▷ Train vector field v_θ in BC flow policy π_θ
  $x^0 \sim \mathcal{N}(0, I_d)$
  $x^1 \leftarrow a$
  $t \sim \text{Unif}([0, 1])$
  $x^t \leftarrow (1-t)x^0 + tx^1$
  Update θ to minimize $\mathbb{E}[\|v_\theta(t, s, x^t) - (x^1 - x^0)\|_2^2]$
  ▷ Train one-step policy π_ω
  $z \sim \mathcal{N}(0, I_d)$
  $a^\pi \leftarrow \mu_\omega(s, z)$
  Update ω to minimize $\mathbb{E}[-Q_\phi(s, a^\pi) + \alpha\|a^\pi - \mu_\theta(s, z)\|_2^2]$
return One-step policy π_ω

Figure: FQL

Algorithm 1: Guided Flow Policy (GFP)

1 **function** *Integrate* $\mu_\omega(s, z)$
  // Explicit discrete Euler integration with M steps
2 **for** $t = 0, 1, \ldots, M-1$ **do**
3 $z \leftarrow z + \frac{1}{M} v_\omega(t/M, s, z)$
4 **return** z
5 **while** *not converged* **do**
6 Sample $\{(s, a, r, s')\} \sim \mathcal{D}$
  // Step 1 -- Train critic Q_ϕ
7 $z' \sim \mathcal{N}(0, I_d)$, $a' = \mu_{\bar{\phi}}(s', z')$
8 Update ϕ to minimize $\mathbb{E}[(Q_\phi(s, a) - r - \gamma Q_{\bar{\phi}}(s', a'))^2]$
  // Step 2 -- Train the distilled one-step actor π_θ
9 $z \sim \mathcal{N}(0, I_d)$, $a^{\pi_\theta} = \mu_\theta(s, z)$,
10 $a^{\pi_\omega} = \mu_\omega(s, z)$ // Using the Integrate-μ_ω function, Line. 1
11 Compute $\lambda = \frac{1}{\frac{1}{N}\sum |Q_\phi(s, a^{\pi_\theta})|}$ // Stop gradient a^{π_θ}
12 Update θ to minimize $\mathbb{E}[-\lambda Q_\phi(s, a^{\pi_\theta}) + \alpha\|a^{\pi_\theta} - a^{\pi_\omega}\|_2^2]$
  // Step 3 -- Train the value-aware BC policy π_ω
13 Compute $g_\eta(s, a) = \frac{\exp\left(\frac{1}{\beta} Q_\phi(s, a)\right)}{\exp\left(\frac{1}{\beta} Q_\phi(s, a^{\pi_\theta})\right) + \exp\left(\frac{1}{\beta} Q_\phi(s, a)\right)}$ // Stop gradient a^{π_θ}
14 $a_t = (1-t)\epsilon + ta$, with $\epsilon \sim \mathcal{N}(0, I_d)$ and $t \sim \mathcal{U}([0, 1])$
15 Update ω to minimize $\mathbb{E}\left[g_\eta(s, a)\|v_\omega(t, s, a_t) - (a - \epsilon)\|_2^2\right]$
Output: $\pi_\theta, \pi_\omega, Q_\phi$

Figure: GFP

Task Category	Offline RL algorithms				
	IQL	ReBRAC	FQL	GFP actor π_β	GFP VaBC π_ω
OGBench antmaze-large-navigate-singletask (5 tasks)	*53* ± 3	**95.9** ± 0.4	88.1 ± 3.4	**93.8** ± 1.5	90.0 ± 1.3
OGBench antmaze-large-stitch-singletask (5 tasks)	30.4 ± 3.2	**89.2** ± 6.6	58.1 ± 8.7	68.9 ± 0.8	57.6 ± 3.2
OGBench antmaze-large-explore-singletask (5 tasks)	12.9 ± 1.7	82.7 ± 7.6	**87.9** ± 6.6	**91.9** ± 0.9	**89.3** ± 1.1
OGBench antmaze-giant-navigate-singletask (5 tasks)	*4* ± 1	**33.2** ± 5.7	16.3 ± 8.2	27.9 ± 8.5	0.8 ± 0.2
OGBench humanoidmaze-medium-navigate-singletask (5 tasks)	*33* ± 2	59.2 ± 12.1	*58* ± 5	**72.0** ± 2.8	35.9 ± 2.7
OGBench humanoidmaze-medium-stitch-singletask (5 tasks)	27.3 ± 2.9	61.1 ± 8.2	**63.2** ± 6.7	**66.2** ± 5.7	39.5 ± 2.1
OGBench humanoidmaze-large-navigate-singletask (5 tasks)	*2* ± 1	12.9 ± 4.2	6.5 ± 2.7	**17.8** ± 9.6	2.4 ± 1.1
OGBench antsoccer-arena-navigate-singletask (5 tasks)	*8* ± 2	55.9 ± 1.5	**60** ± 4	**57.9** ± 1.9	10.3 ± 0.7
OGBench antsoccer-arena-stitch-singletask (5 tasks)	2.8 ± 1.0	22.0 ± 1.5	28.6 ± 2.3	**30.5** ± 2.2	1.4 ± 0.3
OGBench cube-single-play-singletask (5 tasks)	*83* ± 3	*91* ± 2	***96*** ± 1	**98.8** ± 0.4	39.7 ± 4.1
OGBench cube-single-noisy-singletask (5 tasks)	53.2 ± 4.1	**98.4** ± 0.6	**100.0** ± 0.0	**100.0** ± 0.0	**99.9** ± 0.1
OGBench cube-double-play-singletask (5 tasks)	7 ± 1	12.6 ± 1.8	*29* ± 2	**47.2** ± 1.6	6.4 ± 1.0
OGBench cube-double-noisy-singletask (5 tasks)	4.5 ± 0.8	19.6 ± 2.1	38.2 ± 5.3	**63.1** ± 3.3	9.4 ± 0.8
OGBench cube-triple-play-singletask (5 tasks)	0.1 ± 0.1	2.9 ± 1.2	3.9 ± 1.5	**15.9** ± 2.0	7.6 ± 1.6
OGBench cube-triple-noisy-singletask (5 tasks)	4.8 ± 1.2	5.2 ± 2.9	3.5 ± 1.6	**24.5** ± 2.8	8.6 ± 1.2
OGBench puzzle-3×3-play-singletask (5 tasks)	*9* ± 1	21 ± 1	***30*** ± 1	23.1 ± 2.2	19.2 ± 2.9
OGBench puzzle-4×4-play-singletask (5 tasks)	7 ± 1	17.1 ± 1.3	*17* ± 2	**26.1** ± 2.1	9.5 ± 1.1
OGBench puzzle-4×4-noisy-singletask (5 tasks)	0.1 ± 0.0	1.1 ± 0.3	15.6 ± 1.1	**18.8** ± 1.7	**19.3** ± 1.0
OGBench scene-play-singletask (5 tasks)	*28* ± 1	41.6 ± 3.6	***56*** ± 2	53.5 ± 2.9	**57.6** ± 1.7
OGBench scene-noisy-singletask (5 tasks)	16.0 ± 1.2	39.9 ± 2.6	**59.3** ± 1.4	**57.5** ± 0.9	**58.5** ± 1.0
OGBench visual manipulation (5 tasks)	*42* ± 4	*60* ± 2	***65*** ± 2	**62.8** ± 1.5	–
D4RL antmaze (6 tasks)	*17*	*76.8*	***84*** ± 3	**83.1** ± 2.7	70.2 ± 3.0
D4RL Adroit (12 tasks)	*48*	*59*	*52* ± 1	52.8 ± 1.4	49.6 ± 1.3
Minari Adroit (12 tasks)	–	–	40.6 ± 0.4	**48.3** ± 2.3	**46.1** ± 1.7
Minari hopper (3 tasks)	–	–	79.6 ± 10.3	**91.7** ± 4.5	**91.5** ± 12
Minari halfcheetah (3 tasks)	–	–	97.8 ± 2.0	**109.1** ± 2.0	**103.1** ± 1.8
Minari walker2d (3 tasks)	–	–	**121.7** ± 1.3	**124.5** ± 0.8	**122.2** ± 1.1
Average OGBench (105 tasks)	20.4	43.9	46.7	**53.2**	33.1[2]
Average D4RL (18 tasks)	54.0	**64.8**	**62.1**	**63.0**	56.5
Average Minari (21 tasks)	–	–	65.9	**74.1**	**71.6**