A. Pseudo-code

Algorithm 1 Greedy coupling earth moving effort **lower bound** e^L

Data: Expert demonstrations $Y = \{y_i\}_{j\in[1:n]}$, the task horizon m, Agent π, distance metric d
Initialization: Assigning weight $\frac{1}{n}$ to expert demonstrations as $Y' = \{y_j, u_j\}_{j\in[1:n]}, u_j = \frac{1}{n}$
for $i = 1$ **to** m **do**
  Execute action $a^\pi := \pi(s_i)$ and observe s^π_{i+1}. Denote $x_i = (s^\pi_i, a^\pi_i)$ or $x_i = (s^\pi_i)$
  Assign weight $w^\pi = \frac{1}{m}$, EMD effort $e_i := 0$
  repeat
    $y_k, u := argmin_{(y_k, w^e)'} d(x_i, y_k)$
    if $w^\pi \geq u$ **then**
      $e_i := e_i + u_k d(x_i, y_k)$
      $w^\pi := w^\pi - u_k$
    else
      $e_i := e_i + w^\pi d(x_i, y_k)$
      $w^\pi = 0$
    end if
  until $w^\pi \leq 0$
  e_i is the moving effort e^L_i
end for

Algorithm 2 Greedy coupling earth moving effort e^g

Data: Expert demonstrations $Y = \{y_i\}_{j\in[1:n]}$, the task horizon m, Agent π, distance metric d
Initialization: Assigning weight $\frac{1}{n}$ to expert demonstrations as $Y' = \{y_j, u_j\}_{j\in[1:n]}, u_j = \frac{1}{n}$
for $i = 1$ **to** m **do**
  Execute action $a^\pi := \pi(s_i)$ and observe s^π_{i+1}. Denote $x_i = (s^\pi_i, a^\pi_i)$ or $x_i = (s^\pi_i)$
  Assign weight $w^\pi = \frac{1}{m}$, EMD effort $e_i := 0$
  repeat
    $y_k, u := argmin_{(y_k, w^e)'} d(x_i, y_k)$ *The greedy coupling strategy
    if $w^\pi \geq u$ **then**
      $e_i := e_i + u_k d(x_i, y_k)$
      $w^\pi := w^\pi - u_k$
      $Y'.pop(y_k, u_k)$
    else
      $e_i := e_i + w^\pi d(x_i, y_k)$
      $u_k := u_k - w^\pi$
      $w^\pi = 0$
    end if
  until $w^\pi \leq 0$
  e_i is the moving effort e^g_i
end for

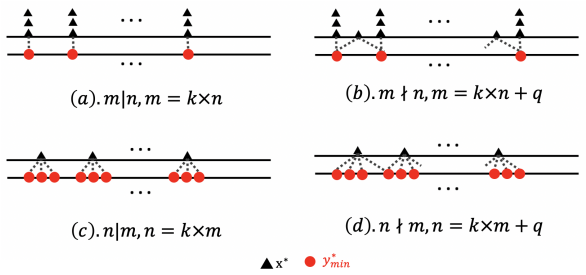

Figure 6. The 4 optimal greedy coupling cases

B. Justification for Proposition 1

Intuitively, when an optimal PWIL policy is acquired, the greedy coupling EMD will be minimized. With the lower bound of greedy EMD earth moving effort defined e^L, we reformulate the Proposition 1 as follows:

Proposition 2. *Given an expert finite distribution set as $Y = (y, u) \in D^{d,n}, u = \frac{1}{n}$, the task's horizon as m, agent generated trajectory set as $X = (x, w) \in D^{d,m}, w = \frac{1}{m}$; Denote the optimal policy as π^*, the obeservation as time step t as x_t^*; the greedy earth moving effort as $e^g(x_t^*)$ and the earth moving effort lower bound as $e^L(x_t^*)$. The following equality holds for any t when $m \mid n : m = k \times n$ or $n|m$: $n = k \times m$, where $k, m, n \in N_+$:*

$$e^g(x_t^*) = e^L(x_t^*) \tag{15}$$

Proof. Given an optimal PWIL imitation policy π^*, the observation generated by π^* at time step t as x_t^*. The optimal policy π^* obtains a finite set X^* that:

$$x_t^* \in X^*, X^* = \arg\min_{X^*} \mathrm{EMD}^g(X, Y)$$

- Fig.6(a): For case $m \mid n, m = kn, k \in N_+$, each hole in the demonstration set Y will be perfectly filled with k piles of dirt, **each pile of dirt x_t^* with mass $\frac{1}{m}$ will be moved entirely to its closest hole $y_{j_{min}}^*$** with capacity $\frac{1}{n}$. For each pile of dirt, its minimum earth moving effort is 0. As is shown in Fig. 6(a): in such case $EMD^*(X^*, Y) = 0$, meaning that the optimal policy conditioned EMD^g is 0. It also indicates that for the optimal policy π^*'s obervation x_t^*:

  $$\forall x_t^*, x_t^* \in Y.$$

  As each pile of dirt's position is identical to its closest hole's position and each hole is able to be filled with k piles of dirt, so that $e^g(x_t^*) = e^L(x_t^*) = 0$.

- Fig.6(c) presents the case $n \mid m, n = km, k \in N_+$. Since the EMD's lower bound is the distance between the centroids of two distributions [1], the EMD's lower bound is also the EMD^g's lower bound. Now we have, for EMD^g, its lower bound is the distance between the centroids of X and Y. This lower bound could be written as:

  $$\mathrm{EMD}^L(X^*, Y) = ||\bar{X}^* - \bar{Y}||,$$

  where $||\bar{X}^* - \bar{Y}||$ is the distance between the centroids of two distribution sets.

  When $n = km, k \in N_+$, the minimized $\mathrm{EMD}^g(X, Y)$ is equal to its lower bound $\mathrm{EMD}^L(X, Y)$: each x_i is equally divided and moved to its k closest holes based on the greedy nature (it could be considered as that the demonstration set Y is clustered into m clusters with the number of k holes, and $x_i^{\pi^*}$ is the cluster centroid). **Each pile of dirt x_i could be equally divided and moved to k holes with mass $\frac{1}{n}$**. In this case:

  $$e^g(x_t^*) = e^L(x_t^*) = \sum_{j=0}^{k} \frac{1}{n} d(x_t^*, y_j).$$

[1] The lower bound of EMD is proven in: Cohen, Scott, and Leonidas Guibas. The earth mover's distance: Lower bounds and invariance under translation. STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1997.

this proposition does not hold anymore for the case $m \nmid n$ or $n \nmid m$. However, we could still have a soft assumption that this proposition also works when $m \nmid n$ or $n \nmid m$. We now discuss these two cases:

- Fig.6(b) presents the minimized EMD^g case when $m \nmid n : m = kn + q, k, q \in N_+, q < n$. It could be considered that firstly there are maximum $k \times n$ piles of dirt that could be completely moved to its closest holes and their positions are identical to their corresponding holes' positions. It means that based on the greedy nature of greedy earth moving strategy, there exist $k \times n$ piles of dirt that will be moved and only moved to their closest holes. For these piles, their positions are identical to their corresponding holes and $e^g(x_t^*) = e^L(x_t^*)$.

  However, there are still q piles of dirt remaining to be moved while these dirt piles cannot be entirely allocated to their closest holes. For these piles of dirt, $e^g(x_t^*) > e^L(x_t^*)$.

  It is worth noting that $k \times n > q$, which indicates that a larger amount of piles can be assigned to its closest holes. Practically we could consider $e^g(x_t^*) \approx e^L(x_t^*)$ when $m = kn + q, k, q \in N_+, q < n$.

- Fig.6(d) presents the minimized EMD^g case when $n \nmid m : n = km + q, k, q \in N_+, q < m$. As the centroid distance is the lower bound of EMD^g, when an optimal policy π^* (EMD^g is minimized) is acquired, the shape of trajectory set X will have similar properties as when $n = km$. In this case, each dirt will be nearly the centroids of expert set Y. Each observation x_t^* will be divided and moved to at least $k + 1$ holes. We have a soft assumption that $e^g(x_t^*) \approx e^L(x_t^*)$.

□

In summary:

- if $m \mid n : m = k \times n, k, m, n \in N_+$, $\text{EMD}^g(F^g, X, Y) = 0$, each hole $y_j \in Y$ will be filled with k piles $x_i \in X$ and x_i's position is identical to y_j, $e^g(x_t^*) = e^L(x_t^*)$.
- if $n \mid m : n = k \times m, k, m, n \in N_+$, each x_i is the centroid of k holes y_j, and x_i will be matched with its closest k holes y_j. $e^g(x_t^*) = e^L(x_t^*)$.
- if $n \nmid m$ or $m \nmid n$, practically we could consider $e^g(x_t^*) \approx e^L(x_t^*)$.

C. Experiments details

For PWIL (greedy coupling EMD) reward calculation, we use the source code provided by PWIL. We choose TD3 (Fujimoto et al., 2018) as the representative of off-policy RL. The implementation is based on stable-baselines3[2] (Raffin et al., 2019). We used 10 random seeds for each experiment.

C.1. Demonstration generation

We first train the expert policies based on the TD3 implementations provided in stable-baselines3 using its default parameters (Raffin et al., 2019). After training, we use the trained model to run Pybullet tasks. After training, we generate 20 expert trajectories with a 20 random seeds for each PyBullet task. The trajectory's horizon is 1000 steps for all tasks.

C.2. PWIL setups

For PWIL reward calculation, we used the source code provided by the original PWIL work. The script can be found on PWIL's official Github page[3]. The reward function is implemented the same as in the original PWIL work (**?**):

$$r_i = R(e_i^g) = 5 \times (\frac{T}{\sqrt{|S| + |A|}}) \times e_i^g. \tag{16}$$

$\frac{T}{\sqrt{|S|+|A|}}$ is a scaling factor that acts as a normalizer on the dimensionality of the state and action spaces and on the time horizon of the task. We use the standardized Euclidean distance which is the L2 distance on the concatenation of the observation and the action, weighted along each dimension by the inverse standard deviation of the expert demonstrations.

[2]https://github.com/DLR-RM/stable-baselines3

[3]https://github.com/google-research/google-research/tree/master/pwil

C.3. Network structures and hyperparameters

The actor architecture is a 3-layer neural network (the default setup in stable-baselines3): the first layer has size 400 with tanh activation and layer normalization (Ba et al., 2016), the second layer and third layer have size 300 with ReLU-activation (Agarap, 2018), the last layer size is equal to the action space dimension with a tanh activation scaled to the action range of the environment. For the critic network we use a 3-layer neural network: the first layer has size 400 with tanh activation and layer normalization, the second layer is of size 300 with ReLU activation, the third layer is of size 256 with ReLU activation, the last FC layer is of dimension 1 to output the value (for TD3). For exploration, we use a Gaussian noise layer on top of the last layer with standard deviation $\sigma = 0.1$, and clip to the action range of the environment. We evaluate the agent without exploration noise.

We chose $\gamma = 0.99$ for reward discounted accumulative calculation; replay buffer size=10e6; the soft update coefficient $\tau = 0.005$; 1000 steps of the model to collect transitions for before learning starts (all parameter is the stable-baselines3 default parameters). All these parameters (network architectures, hyperparameters) use default values of stable-baselines3.

We use the Adam optimizer (Kingma & Ba, 2014) with $\lambda a = 5105$ for the actor and $\lambda c = 7105$ for the critic. We use a batch size of 256. We prefill the replay buffer with 10000 state-action pairs from the set of demonstrations (which means that we put multiple times the same expert transitions in the buffer). We perform updates on the actor and the critic every $k = 1$ interactions with the environment.

C.4. BC, SQIL, AdRIL setups

We compared PWIL and our variants with some off-policy imitation learning methods: BC, SQIL, and AdRIL. For these experiments, we used the source code provided from work (Swamy et al., 2021b;a) [4]. For policy and value networks' structures, we modified them the same as in our PWIL implementations. For exploration, we use a Gaussian noise layer on top of the last layer with standard deviation σ= 0.1, and clip to the action range of the environment.

More specifically, SQIL and AdRIL is built on top of the Stable Baselines implementation of SAC. AdVIL is written in pure PyTorch.The rest of the hyper-parameters of reinforcement learning training remain the same as the our PWIL implementations. For behavior cloning baselines, we used the hyper-parameters from AdRIL's experiments (Swamy et al., 2021b;a).

D. Additional Experiments

We provide the additional experiments to prove the findings in Proposition 2: when the optimal policy π^* is acquired, $e^L = e^g$, which makes $r^* = r^U$, no matter what value m or n is. It also indirectly indicates that $\mathcal{B}^U(Q)$ is equal to the true Bellman operator $\mathcal{B}^*(Q)$ when the optimal policy is acquired. The additional $\mathcal{B}^U(Q)$ shall improve the performance no matter what value m or n is.

For the expert demonstration set, there are n holes to be filled; for the policy trajectory set, there are m piles of dirt to be moved. To verify this finding, we create this experiment for different cases when $m \neq n$: for expert demonstrations, there are 5 expert trajectories given, each trajectory contains 1000 steps, thus $n = 5000$. For policy task horizon, there are 1000 steps, thus $m = 1000$. We choose 3 different subsampling ratios for expert demonstration:

- we do not subsample the expert trajectory, subsample rate=1, in this case $m = 1000, n = 5000, n|m$.
- we subsample the expert trajectory every 13 steps (subsample rate=13), in this case $m = 1000, n = 384, m \nmid n$.
- we subsample the expert trajectory every 20 steps (subsample rate=20), in this case $m = 1000, n = 250, m|n$.

Table 3 shows the policy evaluation results of the original PWIL and PWIL-upper (PWIL with $\mathcal{B}^U(Q)$). **Note that the accumulative reward is based on PWIL reward in this experiment instead of the true environment reward.**

We use t-test with p-value threshold of 0.05 to verify if the performances have a significant difference. In general, for different demonstration subsampling ratios, the PWIL-upper consistently outperforms the original PWIL: **1)** when the subsample rate is 1, the PWIL-upper variant performs better in tasks Hopper, Ant and HalfCheetah **2)** when the subsample

[4] https://github.com/gkswamy98/pillbox

Table 3. Additional justification for Eq. 10: In the case $n \nmid m$ ($k = 13$), PWIL-upper is able to improve the task performance the same as the cases $n \mid m$ (k=1) and $m \mid n$ (k=20). In each subsampling setup, PWIL-upper entails performance gains for most of the tasks. The threshold of T-test is chosen when $\text{p-value} < 0.05$. Our upper-bounded PWIL performance outperms the orginal PWIL in most of the case

5 Demonstrations	Subsample 1 ($n \mid m$)			Subsample 13 ($n \nmid m$)			Subsample 20 ($m \mid n$)		
PWIL reward Evaluation	PWIL(baseline) (mean±std)	PWIL-upper (mean±std)	p-value	PWIL(baseline) (mean±std)	PWIL-upper (mean±std)	p-value	PWIL(baseline) (mean±std)	PWIL-upper (mean±std)	p-value
Humanoid	288.3 ± 75.1	285.2 ± 74.3	6.9e-01	223.4 ± 64.7	$\mathbf{248.1 \pm 70.4}$	4.0e-04	189.8 ± 93.3	$\mathbf{236.5 \pm 70.0}$	6.6e-08
Walker2D	1402.4 ± 82.1	1367.8 ± 364.3	2.0e-01	1169.6 ± 214.0	$\mathbf{1221.1 \pm 145.2}$	6.5e-03	986.9 ± 301.3	$\mathbf{1150.4 \pm 215.9}$	3.2e-09
Hopper	1969.4 ± 147.6	$\mathbf{2062.0 \pm 126.6}$	1.9e-10	1706.1 ± 125.0	$\mathbf{1803.5 \pm 92.6}$	2.1e-16	1672.5 ± 108.6	$\mathbf{1758.6 \pm 110.1}$	1.6e-13
Ant	1090.4 ± 446.2	$\mathbf{1214.4 \pm 111.2}$	2.0e-04	887.1 ± 445.2	$\mathbf{1057.0 \pm 334.3}$	3.4e-05	549.0 ± 458.3	$\mathbf{943.1 \pm 317.9}$	4.6e-20
HalfCheetah	1795.9 ± 210.3	$\mathbf{1862.6 \pm 107.2}$	1.0e-04	1490.7 ± 71.7	1478.1 ± 78.4	1.1e-01	1448.0 ± 98.2	$\mathbf{1489.8 \pm 53.6}$	4.4e-07

rate is 13, except task HalfCheetah, PWIL-upper significantly outperforms the original PWIL. **3)** when the subsample rate is 20, the PWIL-upper beats the original PWIL in all tasks.

This experiment proves that $\mathcal{B}^U(Q)$ is effective in various conditions when the number of expert observations m does not match the task's horizon n. It also indirectly verifies our Claim. 2: when the optimal policy π^* is acquired, $e^g(x_t^*) = e^L(x_t^*)$, and it makes $r^* = r^U$, $\mathcal{B}^U(Q) = \mathcal{B}^*(Q)$.