1) Minimizing Inverse Dynamics Disagreement

This paper proves that the gap in LFD and LFO methods lies in the disagreement of the inverse dynamics models of the imitator and the expert and the upper bound of this gap is considered as a negative causal entropy of the state occupancy measure which can be minimized in a model free way. We are modeling LFD using GAIL and LFO using GAIfO. This term contains the mutual information term which can be optimized using methods such as MINE. The final loss is a combination of the term from naive LFO, and the entropy and MI term for bridging the gap. The first term can be realized using a GAN like approach, using a discriminator D and a policy network pi. The loss can be used to update the policy pi. The training method is similar to GAIL, but uses a state action pair.

2) To Follow or Not to Follow

This method selects reachable states first and then learns how to reach the selected states, instead of the imitator following each state the expert takes. This is an implementation of hierarchical RL, where there are two policies, one meta policy and one low level policy. Basically the meta policy selects what state to target, which is the sub goal and the low level policy, taking the subgoal as the target plans how to reach that selected state. After this step, the meta policy again picks the next subgoal and the process repeats. We consider the meta policy as pi(g|ot, tau; theta). One og is chosen, the low level policy generates an action at ~ pi_low(o_t, o_gt; phi) and generates a rollout until the agent reaches the subgoal or episode ends. The meta policy gets a +1 reward every time the subgoal is reached (|o_gt - o_(t+1)| < e). The low level policy and meta policy are jointly trained used buffers R_meta, and R_low.

3) Imitation Learning as State Matching via Differentiable Physics

This paper mainly focuses on avoiding a double loop as it is in IRL methods, which includes learning a reward function and a subsequent policy. The loss used for policy learning includes the terms for Deviation loss and a Coverage loss., whose linear combination defines the alpha-Chamfer loss. The crux here is to use the differentiable dynamics as a physics prior and incorporate it into the computational graph for policy learning. We don’t use the L2 loss to relax the enforcement of the exact matches of the states.

4) Plan your State and Learn Skills

Similar to 2), this decouples a policy as a high level state planner and an inverse dynamics model. The IDM can be trained online by minimizing the KL divergence between the IDM of the agent policy and the sampling policy. The decoupling can be referred to as pi = (tau_pi)-1 (tau(pi_e)). The state planner can be trained by minimizing the divergence of the state occupancy measure. The policy gradient term for the high level planner can be formulated using the above terms.

5) Self Supervised Adversarial Imitation Learning

SAIL is an IL method with the intersection of self supervised learning and adversarial methods. It describes 4 models : i) M (the IDM here) P(a|s, s') ii) policy pi P(a|s), iii) generative model G P(s' | s, a) and iv) discriminator D. M is trained using supervised learning with tuple generated by an agent. With inferring actions, we can train pi with behaviour cloning. G is also updated during this training. We will append all the samples of the tuple that D could not differentiate between imitator and expert, and also make an update to the policy based on the behaviour of D. The policy is updated by the gradient flow from D. G acts as a forward dynamics model and helps in the following ways :

6) Extrapolating beyond Suboptimal Demonstrations

The algorithm here considers a set of demonstrations, with ranked optimalities. It has two steps : reward inference and policy optimization. For reward inference, the reward function is parametrized by a neural network r1 < r2 if t1 is ranked below t2. (r, t are rewards and trajectories respectively) The loss function trains a classifier (OvO), and the probability is represented as a softmax normalized distribution. We thus use the neural network r to act as a reward for the policy optimization.

7) Imitation by Predicting Observations

Here, the imitator model and the demonstrator models are effect models. The demonstrator effect model can be trained using the expert dataset by gradient descent. We sample trajectories using our policy and add them to the replay buffer, and we sample a batch of them to train the imitator effect model. The reward is taken as the difference of log of the two effect models, which is used to train the imitator policy. The imitator effect model can be trained using supervised methods on the sample trajectories.

8) Imitation from Bootstrapped Contrastive Learning

Here we are dealing with only visual observations. As the title suggests the main idea of the paper is dealing with contrastive learning. For agent training, the authors describe two steps, the Alignment Phase and the Interactive Phase. The alignment phase deals with learning an encoding function f_w and an associated distance metric between the agent trajectories. We use a reward based on the distance between the agent and expert encodings to train the RL algorithm. There are specific techniques for image encoding which are used for a frame encoding. The sequence encoding training involves the triplet loss.

9) Imitating Latent Policies from Observation

This algorithm deals with two steps : policy learning in a latent space, and action mapping. A generative model is first trained to predict the next state. This is used to learn the latent forward dynamics on the expert state observations. The ground truth next state can be used for the loss function to train this generative model. The policy can now be trained concurrently on the expert data. The expected value of the next state can be found out by integrating the policy. The loss is then the expected next state and the true next state difference. The network is trained with the sum of the above losses. Action mapping can be done in a supervised manner with the collection of experiences.

10) Generative Adversarial Imitation from Observation

This aims to minimize the difference between the state occupancy measure between the imitator and the expert. The loss function can be solved as the generative adversarial loss, with the discriminator loss over the agent being used to update the policy, and the net loss being used to update the discriminator. The central algorithm is derived using the convex conjugate concept. The entire process can be summarized as bringing the distribution of the imitators' state transitions closer to that of the expert.

11) Zero Shot Visual Imitation

This uses a goal conditioned skill policy, which can be inferred to as the inverse dynamics model. The key difference here is the consideration of multimodality, which says multiple actions can lead to the same future state from the initial observation. To curb this, a forward consistency model is used. A binary classifier model goal recognizer is also used to ensure that the actions inferred from the GSP lead to a final observation close to the ground truth observation. The FCL model penalizes the difference in the next states by predicting the action and the ground truth action. The objective for the FCL includes the term describing the difference between the expert observation and the observation reached by following the GSP, and also the difference between the expert observation and the observation reached by the forward dynamics model. The same idea can be extended to a multistep nature.

12) State only Imitation Learning for Dexterous Manipulation

This describes using an inverse dynamics model and then jointly updating the policy network with a policy gradient. The inverse dynamics model can be trained in a supervised manner by minimizing the difference between the predicted action and the ground truth action from the sample collected by the agent. The net gradient term consists of a policy gradient term and an auxiliary imitation term which samples (s, a') from the new D' (collecting samples using the IDM).

13) Imitation from Observation : Context Translation

This paper deals with observations derived from an alternative viewpoint. A translation model is first trained on pairs of observations derived from different contexts. Consider two different demonstrations D_1 and D_2, with contexts w_1, and w_2 respectively, this model looks at the first observation from D_2 and predicts the future observations in D_1 (The assumption is that these are aligned in time). This model is formally M(o_ti, o_0j) which outputs (o_tj). There are four components here : 2 encoders, 1 translator and 1 decoder. The translator is the naive M, and the loss is the difference between the o_jt. There is also a reconstruction loss derived from the autoencoder architecture. Finally, there is an alignment loss between the translated output z_3 and Enc(o_tj). The combined loss can be used to train the entire translation model. We also need to track the features, to know the actions to be taken in the target context. The feature tracking has 2 rewards, which penalizes the difference in the encoding and the translation function. There is also a weak image reward that penalizes the policy for experiencing observations that differ from the translated observations.

14) State Alignment Based Imitation Learning

This paper deals with the problem of transition dynamics mismatch. The integral components are deviation correction, global state alignment and regularized policy update. For local alignment, a VAE is used to predict the next state, and an inverse dynamics model is used to generate the actions. The global alignment is done using Wasserstein distance to solve the problem of imitators going far off the expert distribution. The discriminator is trained using the gradient penalty term. Finally the policy is updated using losses from the local alignment and global alignment steps. The KL Divergence term acts as the regularizer for the local alignment step.

15) Behaviour Cloning from Observations

The idea here is to train an inverse dynamics model using the sample of trajectories by the agent policy. We then infer the actions from the expert set of trajectories using this trained inverse dynamics model. This now creates a dataset of {state, action} tuple, after which we can use Behaviour Cloning (supervised approach) to improve the policy. Note the use of samples in the pre demonstration part and the post demonstrations part.

16) State only Imitation with Transition Dynamics Mismatch

This is termed as indirect imitation learning since we are using a surrogate policy to imitate. We first derive a lower bound on the max entropy irl problem, since we are not provided with actions. The next step is to use AIRL to reduce the distance in the distributions between the state occupancy of the imitator and the one induced by this surrogate policy. We then use WGANs to reduce the distance between the state occupancy between the expert demonstrations and this surrogate policy, hence optimizing it and bringing it closer to the true expert. The surrogate policy trajectories are added to a buffer based on a priority based protocol,since we are not parametrizing this surrogate policy. The Wasserstein critic is thus updated using this distribution.

17) MobILE : Model Based State Only Imitations

The core idea of this paper revolves around the imitation v/s exploration tradeoff. There are 4 components central to this paper : Forward Dynamics Models, discriminator based on integral probability distributions matching, a bonus parameterization and policy. The bonus is more for where the forward dynamics model is unsure of, encouraging exploration. The bonus can be referred to as the uncertainty which is the maximum disagreement of functions in a version space. We are also incorporating the bonus in the discriminator for similar reasons as the tradeoff.

18) Versatile Skill Control via Self-supervised Adversarial Imitation of Unlabeled Mixed Motions

This paper deals with imitating across diverse, unlabeled skills. Two discriminators are used : namely the skill discriminator that differentiates between the skills and the imitator discriminator that differentiates between the imitator and the expert. The imitator discriminator is trained using the LSGAN loss. We next sample a skill z in the beginning of the trajectory on which the policy is conditioned on, along with the state. This unsupervised skill discovery is derived by maximizing the mutual information between the latent skill and the resulting trajectories. The joint optimization of the policy and imitation discriminators is based on adversarial training whereas the joint training of the policy and the skill discriminator is based on a cooperative game.