Supplementary: Object-centric Forward Modeling for Model Predictive Control

Yufei Ye[1] **Dhiraj Gandhi** [2] **Abhinav Gupta**[12] **Shubham Tulsiani**[2]
[1]Carnegie Mellon University [2]Facebook AI Research
yufeiy2@cs.cmu.edu {dhirajgandhi,gabhinav,shubhtuls}@fb.com
https://judyye.github.io/ocmpc/

Appendix A Real Robot pushing

- Robot Setup:To collect real world data we use Sawyer robot. We place a table in front of it where the objects are placed in order for robot to push it. Kinect V2 camera is rigidly attached overlooking the table for RGB-D perception data. The camera is localized with respect to the robot base via calibration procedure.
- Data Collection Procedure: Given the image I_s of table with object on it, we first perform the background subtraction to get the binary mask corresponding to objects . Using this binary mask, we sample a pixel P_m^C which lies on the object. We treat P_m^C as the mid-point of push. For push start pixel P_s^C, we sample pixel around P_m^C in square such a way that it does not lie on top of the object. The end point of the push P_e^C is calculated based on P_s^C and P_e^C. These pixel location P_s^C, P_e^C in image space are converted to corresponding 3D points P_s^R, P_e^R in robot space using the depth image and camera matrix. Then we use off-the-shelf-planner to move robot gripper finger from $P_s^R \rightarrow P_e^R$. The image I_e is recorded after the arm retracts back. For every push we record the tuple of (I_s, I_e, P_s^R, P_e^R). Figure **??** shows some of the pushing data point collected on real robot. In all we have collected 5K pushing data-points on 8 objects.
- Push novel object: To see how well our method generalizes to novel object, we tested it out for pushing measuring tape. In figure 1 blue arrow shows the push predicted by our method to move it to desired location. Even though our method hasn't seen this object during training of forward model, it is able to push it very close to goal location.
- Flip the object location: To test the effectiveness of our method, we tested it on a bit more challenging scenario. In this case, we have 2 objects on the table. The goal configuration is generated by interchanging the position of objects in start configuration. Figure 2 shows the sequence of action taken by our method to carry out this task.

Appendix B Baseline Model Details.

- Implicit forward model (**Imp-Inv**) [**?**]: the model predicts in a implicit feature space where the entire frame is encoded as one implicit feature without further factoring to objects. An inverse model is trained to take in current and goal feature and outputs the action. In testing, the inverse model are applied iteratively to greedily generate action sequence. The inverse model also regularize the forward model to prevent trivial solution.
- Implicit forward model with pixel reconstruction (**Imp-Plan**): The baseline is a variant of Imp-Inv. The action sequences are generated by a planner in the learned feature space. To further regularize the forward model such that it learns a more informative feature space, we train an decoder to reconstruct the frame in pixels. The learned representation of the frame is used by the planner.
- Flow-based prediction model SNA [**?**] (**Flow**): the model learns to predict transformation kernel to reconstruct future frame. In planning, the predicted transformations are applied to designated pixels (location) to estimate their motion.

3rd Conference on Robot Learning (CoRL 2019), Osaka, Japan.

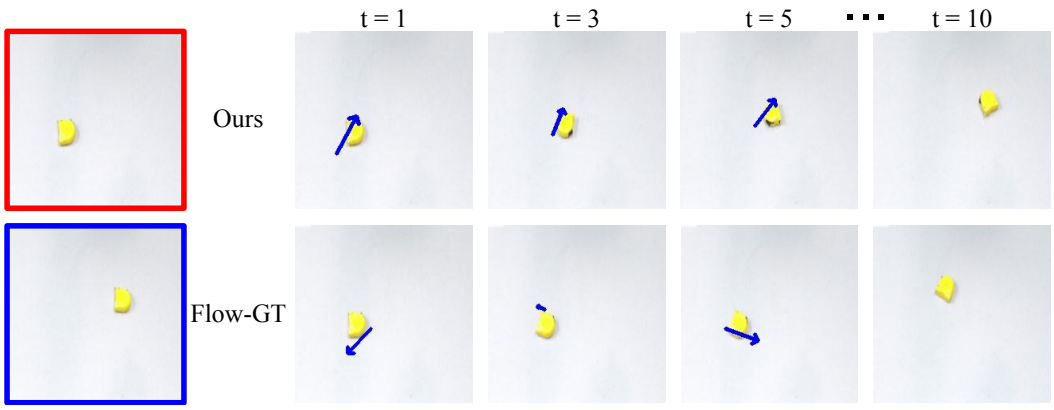

Figure 1: Blue arrow line shows the sequence of action taken by the robot to move objects from start configuration, shown in the red box, to a goal configuration, shown in the blue box.

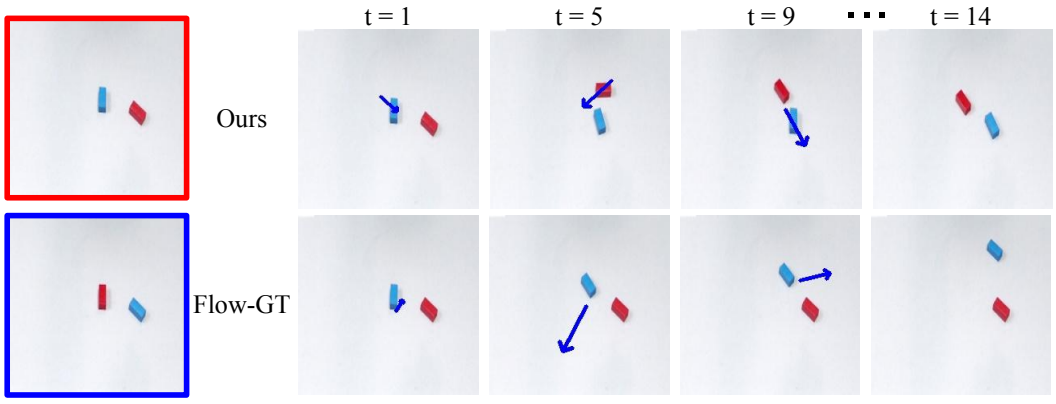

Figure 2: Blue arrow line shows the sequence of action taken by the robot to move objects from start configuration, shown in the red box, to a goal configuration, shown in the blue box.

- Flow baseline with supervision (**Flow-GT**): The original flow baseline only trains with videos in the unsupervised manner. To leverage the additional information, we provide its variant – besides transforming the pixels, the model also transforms the ground truth location $\{\hat{b}_n^t\}$ to $\{b_n^{t+1}\}$ and minimizes the expected distance of transformed location to the ground truth $\{\hat{b}_n^{t+1}\}$.
- **Analytic** baseline. To leverage the location information, a simple analytic solution is to greedily push in the direction of current goal position to desired position. It assume a simple dynamic – the predicted location at the next step is calculated as the delta position of the gripper.

Appendix C Plan with Oracle Location

In this part we compare models when we have access to the ground truth location for each new observation at every time step. After every push, the distance between the current location and the goal in world coordinate is plotted in Figure 3. The analytic baseline should converge to zero because the exact center of mass is given by oracle at every time step, hence serves as ceiling performance. The Imp-Inv barely generalizes to scenarios when the distance of goal and current observation is much ($\times 15$) farther than that in training set. Imp-Plan degenerates in 2 blocks settings, suggesting one feature for the whole frame cannot encode complicated scenes very well. Flow works better than Imp-Plan because the motion space is more tractable. Its performance improves in Flow-GT to leverage location information. Our model outperforms other learning-based methods and performs

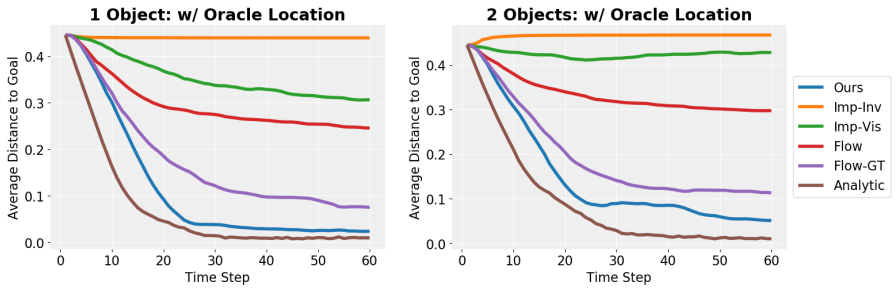

Figure 3: Distance to goal with access to ground truth location at every time step.

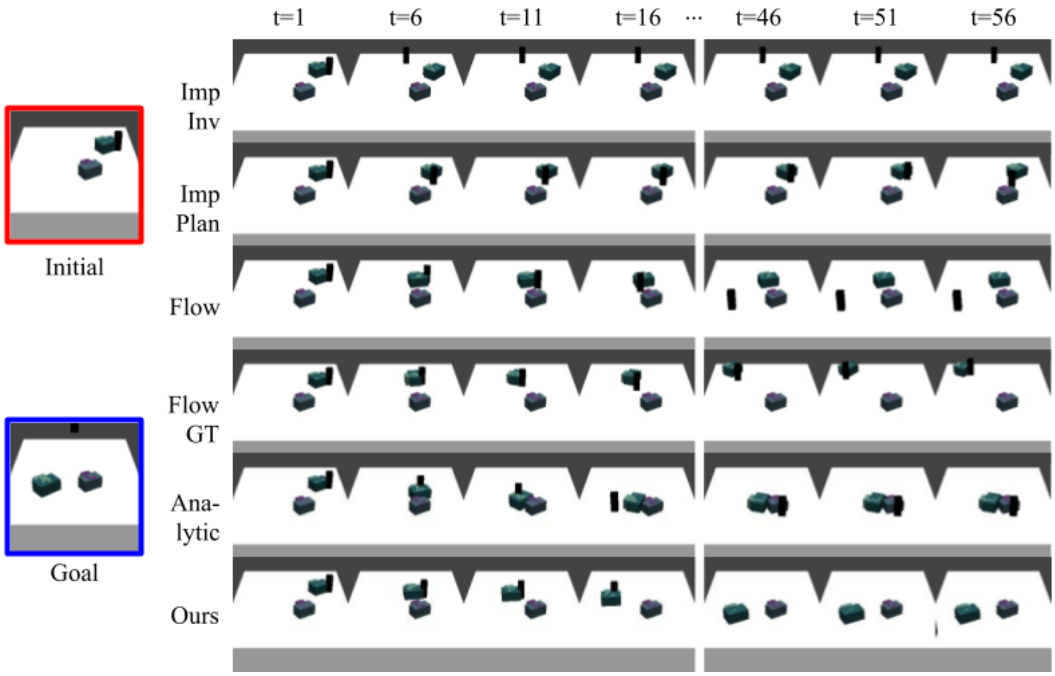

Figure 4: Visualizing an executed action sequence in simulation: Given the initial configuration (in red box) and the goal configuration (in blue box), figure shows the effect of the action predicted by various methods at different time steps. Please refer to the appendix to view all baselines.

comparably to the ceiling performance (Analytic) without manually specifying pushing toward goal through mass center.

Appendix D Qualitative Results of All Baselines.

In this part we show qualitative results in comparison of all baselines. This supplements Figure **??**, which only showcases strong but partial baselines. For more results, please refer to project page.