Ablation	PSNR (↑)	SSIM (↑)
POI, deterministic action prediction	23.59 ± 0.10	0.808 ± 0.004
POI, fixed unit-Gaussian prior	23.06 ± 0.04	0.799 ± 0.002
POI, fully-shared learned prior	23.36 ± 0.05	0.807 ± 0.002
POI, fully-separate learned prior	23.56 ± 0.05	0.808 ± 0.002
POI	$\mathbf{23.79 \pm 0.12}$	$\mathbf{0.813 \pm 0.005}$

Table 4: We compare ablations of the latent variable representation in our model, tested on prediction in the robot manipulation task. Our full model, which uses a stochastic latent action representation composed of components with shared and separate priors, outperforms the ablated variants.

6 Ablation of the action representation

Ablations of the action representation We additionally compare several ablations of the action representation in our model. In the deterministic variant of our method, an inverse model is trained to directly predict the true actions, which corresponds to an instantiation of our graphical model where the latent action z is equal to the action **a**. We also investigate different priors on the stochastic action representation, including a unit Gaussian prior, a fully shared learned prior, where $\mathrm{z} = \mathrm{z}^{shared}$, a fully separate learned prior, where $\mathrm{z} = \mathrm{z}^{domain}$, and our final method, a prior that contains both shared and separate learned components, where $\mathrm{z} = \left(\mathrm{z}^{shared}, \mathrm{z}^{domain}\right)$.

The deterministic variant of our method generally achieves reasonable results. However, the stochastic action representation in our full method performs better, since it does not have to shoehorn the human's actions into the full action space of the robot, but instead can maintain some uncertainty.

The ablation with the fixed unit-Gaussian prior performs poorly because the prior prevents the action encoding from being sufficiently expressive. The fully shared prior does not account for the domain shift between the human and robot data, instead forcing both domains to use the same prior. The fully-separate prior allows for the model to represent the differences between the domains, but it doesn't exploit the similarity between the domains as well as the full POI model, which incorporates both a shared and a separate component of the latent space.

7 Full Architecture

The full architecture of our model is presented in Figure 10.

8 Model Hyperparameters

We selected our hyperparameters through cross-validation. The hyperparameters that are shared between the domains are described in Table 5. The hyperparameters that are specific to the robotic manipulation domain are described

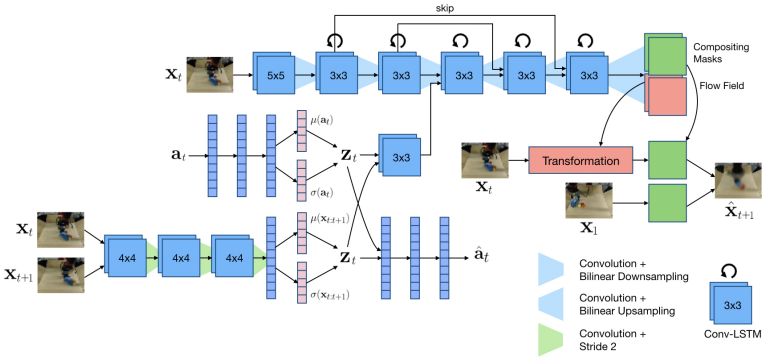


Fig. 10: Our full architecture for learning from observation and interaction data. Our model is composed of the action encoder, inverse model, action decoder, and transition model. The action encoder and inverse model output distributions over $\mathbf{z}_t$ conditioned on the action $\mathbf{a}_t$ and image pair $\mathbf{x}_t$

in Table 6. The hyperparameters that are specific to the driving domain are described in Table 7.

9 Additional Ablations

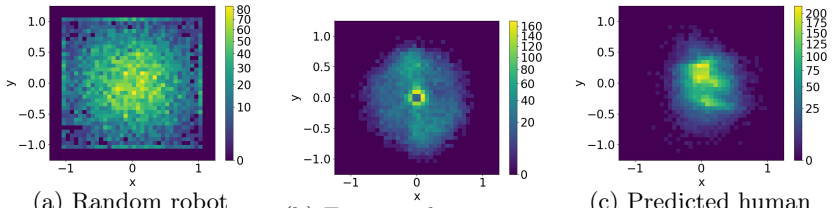
We compare our model to SAVP [34] where the human actions were assumed to be fixed or random. The results for this ablation are shown in Table 8. The prediction model trained only on the robot data outperformed the models trained with human data with incorrect actions, showing that naive action approximations are insufficient to incorporate the information from human video. Our model, POI, outperformed all models, showing that the human videos contain useful information that can improve prediction performance.

10 Robot Planning and Control Experiments

For the control experiments, each task is set up by placing one potential tool into the scene, as well as 2-3 objects to relocate which are specified to the planner by selecting start and goal pixels. The scenes are set up so that because the robot needs to move multiple objects, it is most effective for it to use the tool during its execution. We present the hyperparameters used for the planner in our robotic control experiments in Table 9.

We additionally present more detailed success metrics of the robotic experiments. The success rate at various distance thresholds is shown in Table 10. Our method requires a smaller threshold to achieve all success rates between 20% and 100%.

Figure 11 shows histograms of the random robot actions, the expert robot actions, and the predicted human actions from our model. The predicted human



(a) Random robot actions (b) Expert robot actions (c) Predicted human actions

Fig. 11: Histograms of the x and y components of the actions. Since the human data does not have any actions, the displayed actions were generated by our inverse model. The distribution of predicted human actions of tool-use resembles that of the expert robot actions, suggesting that our model has learned to successfully decode human actions.

actions have a similar distribution to the expert robot actions, even though the expert robot actions were not used to train our model, suggesting that our model mapped the human actions to reasonable locations in the robot's action space.

11 Additional Implementation Details

Additional implementation details are presented in this section.

11.1 Batch Construction

We constructed our batches so that they were made up of a fixed number of examples from each dataset. In all of our experiments, we used a batch size of 12, made up of 9 samples from the interaction data and 3 samples from the observation data.

11.2 Schedule Sampling

In order to improve training, our system initially predicts images from the ground truth previous image. As training continues, the system gradually shifts to using the predicted version of the previous frame.

The probability of sampling an image from the ground truth sequence is given by Equation 9.

$$p = \min\left(\frac{k}{\exp\left(i/k\right)}, 1\right) \tag{9}$$

The iteration number is i, while k is a hyperparameter that controls how many iterations it takes for the system to go from always using the ground truth images to always using the predicted images. This sampling strategy was taken from [34].

12 Domain Shift

Our method of handling domain shift between datasets, described in Section 3.2, is shown in Figure 12.

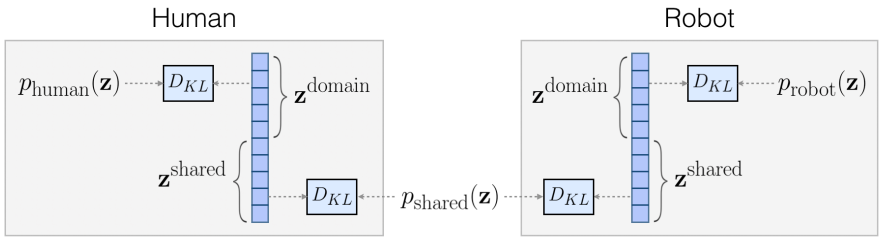


Fig. 12: The partitioned latent space. We partition our latent space $\mathbf{z}$ into two components, $\mathbf{z}^{\text{shared}}$, which captures the parts of the latent action that are shared between domains, and $\mathbf{z}^{\text{domain}}$, which captures the unique parts of the latent action. We enforce this separation by learning the same prior for $\mathbf{z}^{\text{shared}}$ in all domains and a different prior for $\mathbf{z}^{\text{domain}}$ in each domain.

13 Action Visualization

We visualize the histogram of the robot actions in Figure 11. All near-zero actions were removed from the histogram of the expert robot data to remove the long periods of time where the robot is stationary in that dataset.

14 Additional Qualitative Results

Additional qualitative results are presented in this section.

14.1 Video Prediction in the Driving Domain

A version of Figure 4 with more images is shown in Figure 13. We also present the sequence that is best for the baseline in Figure 14. We present the sequence that has the median difference between methods in Figure 15.

14.2 Video Prediction in the Robotic Manipulation Domain

A version of Figure 7 with more images is shown in Figure 16. We also present the sequence that is best for the baseline in Figure 17. We present the sequence that has the median difference between methods in Figure 18.

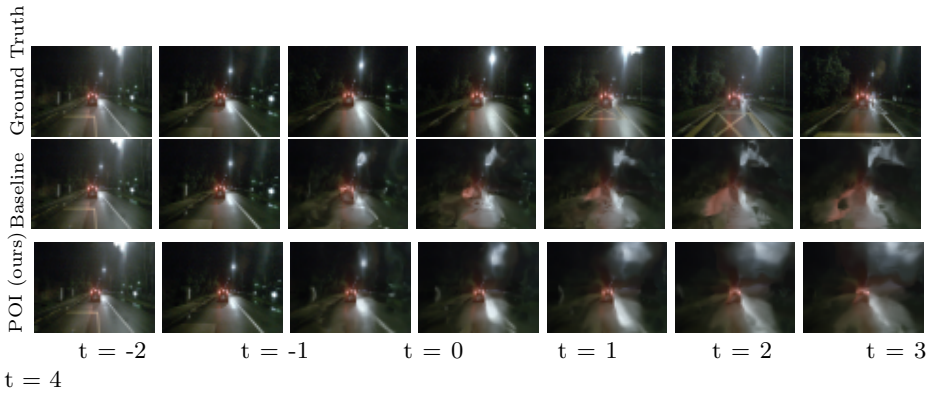


Fig. 13: Example predictions on the Singapore portion of the Nuscenes dataset. This sequence was selected for large MSE difference between the models. We compare our model to the baseline of the SAVP model trained on the Boston data with actions. Our model is able to maintain the shape of the car in front.

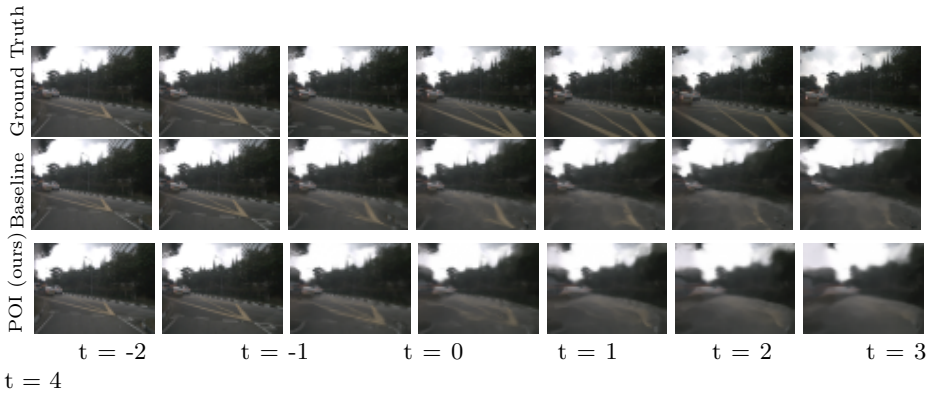


Fig. 14: Example predictions on the Singapore portion of the Nuscenes dataset. This sequence was selected because the baseline had the largest improvement in MSE relative to our model. We compare our model to the baseline of the SAVP model trained on the Boston data with actions. Even in the worse case, our model performs comparably to the baseline model.

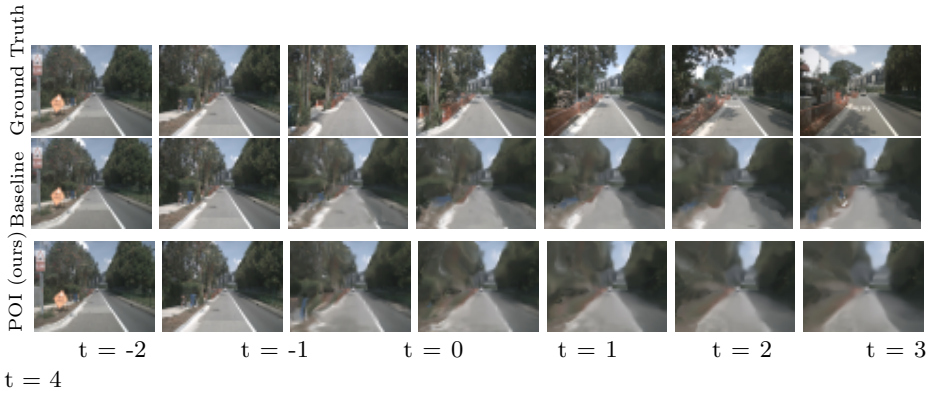


Fig. 15: Example predictions on the Singapore portion of the Nuscenes dataset. This sequence was selected by ordering all of the sequences in the training set by the difference in MSE between the baseline and our model and selecting the middle sequence. We compare our model to the baseline of the SAVP model trained on the Boston data with actions. Even in the worse case, our model performs comparably to the baseline model.

Hyperparameter	Value
Action decoder MSE weight	0.0001
Action encoder KL weight	10^-6
Jensen-Shannon Divergence weight	10^-7
TV weight	0.001
Image L1 reconstruction weight	1.0
Optimizer	Adam [32]
Learning rate	0.001
Beta1	0.9
Beta2	0.999
Schedule sampling k	900
Action encoder channels	64
Action encoder layers	3
Inverse model channels	64
Inverse model layers	3
Generator channels	32

Table 5: Hyperparameter values

Hyperparameter	Value
Dimensionality of $\mathbf{z}^{\text{domain}}$	2
Dimensionality of $\mathbf{z}^{\text{shared}}$	3
Prediction horizon	15

Table 6: Hyperparameter values specific to the robotic manipulation domain

Hyperparameter	Value
Dimensionality of $\mathbf{z}^{\text{domain}}$	0
Dimensionality of $\mathbf{z}^{\text{shared}}$	3
Prediction horizon	5

Table 7: Hyperparameter values specific to the driving domain

Ablation	PSNR (↑)	SSIM (↑)
SAVP w/random robot	23.31 ± 0.10	0.803 ± 0.004
SAVP w/random robot, human w/zero actions	23.21 ± 0.04	0.794 ± 0.002
SAVP w/random robot, human w/gaussian random actions	23.14 ± 0.05	0.796 ± 0.002
POI(ours) w/random robot, human w/no actions	$\mathbf{23.79 \pm 0.12}$	$\mathbf{0.813 \pm 0.005}$

Table 8: Ablations over simple synthetic actions. Our method outperforms training SAVP on the human data with random or zero actions.

Hyperparameter	Value
Robot actions per trajectory	20
Unique robot actions per trajectory	6 (each repeated x3)
CEM iterations	4
CEM candidate actions per iteration	1200
CEM selection fraction	0.05
Prediction horizon	18
Number of goal-designating pixels	3

Table 9: Hyperparameter values specific to the robot control experiments

Distance Threshold	5cm	7cm	10cm	12cm	15cm	17cm	20cm	22cm	25cm
SAVP	16.7%	16.7%	23.3%	40%	86.7%	93.3%	96.7%	96.7%	100%
POI (ours)	10%	16.7%	40%	60%	93.3%	96.7%	100%	100%	100%

Table 10: Success rate at various distance thresholds. Our method requires a smaller threshold to achieve all success rates between 20% and 100%.

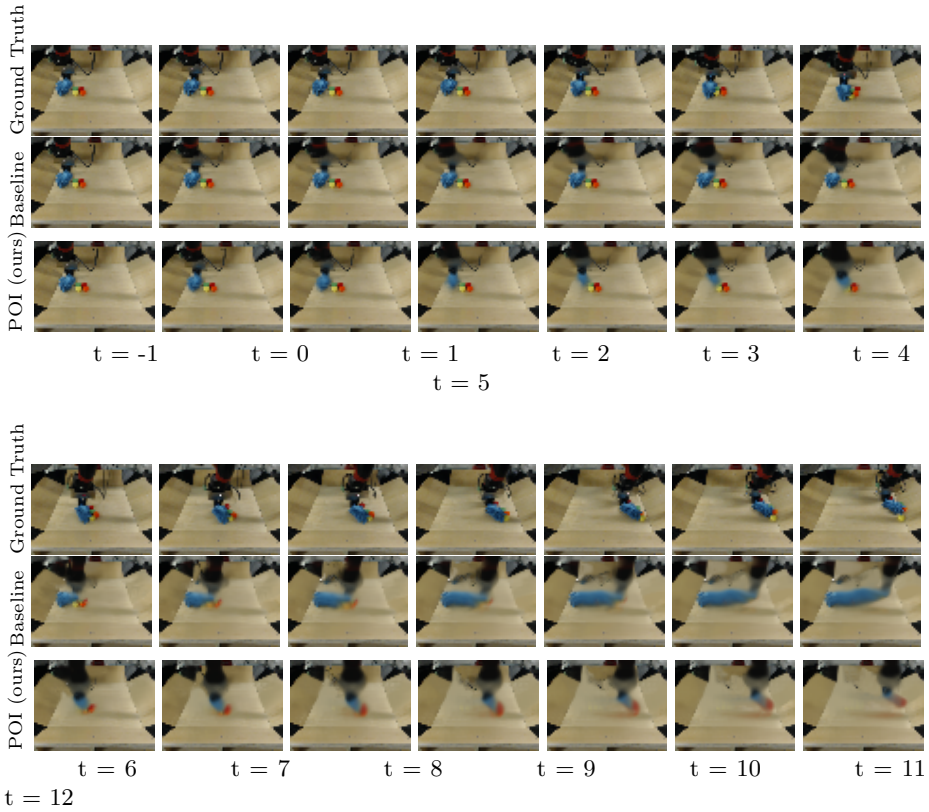


Fig. 16: Example predictions on the robotic dataset. The first image is the context image. We compare our model to the baseline of the SAVP model trained with random robot data. This sequence was selected to maximize the MSE difference between the models. Our model more accurately predicts both the tool and the object it pushes.

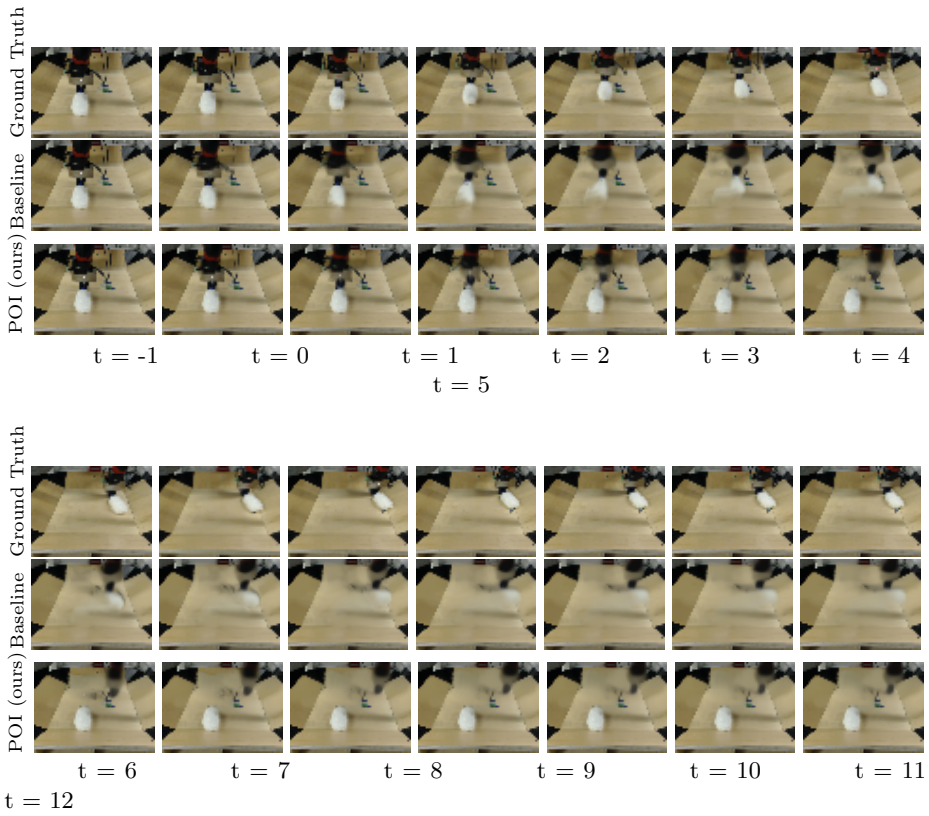


Fig. 17: Example predictions on the robotic dataset. The first image is the context image. We compare our model to the baseline of the SAVP model trained with random robot data. This sequence was selected to maximize so that the baseline had the largest improvement in MSE relative to our model. Our model fails because it was too pessimistic about grasping the narrow handle of the brush.

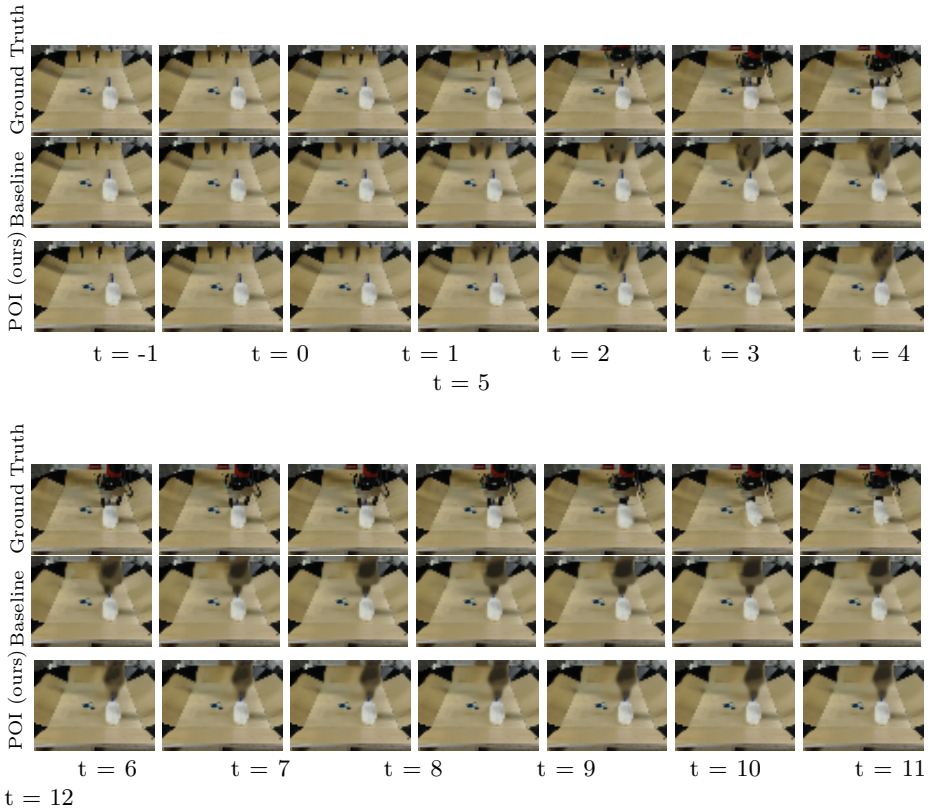


Fig. 18: Example predictions on the robotic dataset. The first image is the context image. We compare our model to the baseline of the SAVP model trained with random robot data. This sequence had the median difference in MSE between our model and the baseline.