A. Discriminative Models: Classifier vs. Energy model

In this section, we assume the dataset as described in VPA, $\mathcal{D} = \{o_1^i, ..., o_{T_i}^i\}_{i=1}^n$. There are two ways of learning a model to distinguish the positive from the negative transitions.

Classifier: As noted above, SPTM first trains a classifier which distinguishes between an image pair that is within h steps apart, and the images that are far apart using random sampling. The classifier is used to localize the current image and find possible next images for planning. In essence, the classifier contains the encoder g_θ that embeds the observation x and the the score function f that takes the embedding of each image and output the logit for a sigmoid function. The binary cross entropy loss of the classifier $L_{SPTM}(\theta, \psi; \mathcal{D})$ is

$$= -\sum_{(z_t, z_{t+k})\sim\mathcal{D}} (\log \frac{f_\psi(z, z_{t+k})}{1 + f_\psi(z_t, z_{t+k})} + \log \frac{1}{1 + f_\psi(z_t, z_t^-)})$$

$$= -\sum_{(z_t, z_{t+1})\sim\mathcal{D}} \log \left[\frac{f_\psi(z_t, z_{t+k})}{f_\psi(z_t, z_{t+k}) + \alpha_\psi^t}\right]$$

where $\alpha_\psi^t = 1 + f_\psi(z_t, z_t^-) + f_\psi(z_t, z_{t+k}) f_\psi(z_t, z_t^-)$, and z_t^- is a random sample from $\mathcal{D}$.

Energy model: Another form of discriminating the the positive transition out of negative transitions is through an energy model. Oord et al. (Oord et al., 2018) learn the embeddings of the current states that are predictive of the future states. Let g be an encoder of the input x and $z = g_\theta(x)$ be the embedding. The loss function can be described as a cross entropy loss of predicting the correct sample from $N+1$ samples which contain 1 positive sample and N negative samples $L_{CPC}(\theta, \psi; \mathcal{D})$ is

$$= -\sum_{(z_t, z_{t+k})\sim\mathcal{D}} \log \left[\frac{f_\psi(z_t, z_{t+k})}{f_\psi(z_t, z_{t+k}) + \sum_{i=1}^N f_\psi(z_t, z_t^{i-})}\right]$$

where $f_\psi(u, v) = \exp(u^T \psi v)$ and $z_t^{1-}, ..., z_t^{N-}$ are the random samples from $\mathcal{D}$.

Note that when the number of negative samples is 1 the loss function resembles the SPTM.

B. Mutual Information (MI)

This quantity measures how much knowing one variable reduces the uncertainty of the other variable. More precisely, the mutual information between two random variables X and Y can be described as

$$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$
$$= \mathbb{E}_{X,Y}\left[\frac{p_{X,Y}}{p_X p_Y}\right].$$

C. Planning as Inference

After training the CPC objective to convergence, we have $f_k(o_{t+k}, o_t) \propto p(o_{t+k}|o_t)/p(o_{t+k})$ (Oord et al., 2018). To estimate $p(o_{t+k}|o_t)/p(o_{t+k})$, we compute the normalizing factor $\sum_{o'\in V} f_k(o', o_t)$ for each o_t by averaging over all nodes in the graph. Therefore, our non-negative weight from o_t to o_{t+k} is defined as $\omega(o_t, o_{t+k}) = \sum_{o'\in V} f_k(o', o_t)/f_k(o_{t+k}, o_t) \approx p(o_{t+k})/p(o_{t+k}|o_t)$.

A shortest-path planning algorithm finds $T, o_0, ..., o_T$ that minimizes $\sum_{t=0}^{T-1} \omega(o_t, o_{t+1})$ such that $o_0 = o_{start}, o_T = o_{goal}$. By Jensen's inequality and the Markovian property of $o_0, ..., o_T$ we have that, $\log \frac{1}{T}\sum_{t=0}^{T-1} \omega(o_t, o_{t+1}) \geq \frac{1}{T}\sum_{t=0}^{T-1} \log \omega(o_t, o_{t+1}) = \frac{1}{T}\sum_{t=0}^{T-1} (\log p(o_{t+1}) - \log p(o_{t+1}|o_t)) = \frac{1}{T}\sum_{t=1}^{T-1} p(o_t) - \log p(o_1, ..., o_{T-1}|o_0 = o_{start}, o_T = o_{goal})$. Thus, since $p(o_t)$ is fixed by uniform asssumption, the shortest path algorithm with proposed weight ω maximizes a lower bound on the trajectory likelihood given the start and goal states. In practice, this leads to a more stable planning approach and yields more feasible plans.

D. Block Insertion Domain

In this domain, we kept the obstacle constant and varied the agent itself. In particular, we uniformly chose from 4 to 10 units, with 6 as the holdout, and then randomly placed those units such that they resembled a contiguous shape. When applying an action, we applied a vertical and horizontal force to the middle block, and also a rotation force on the first and last unit laid down, leading to a total action space of four. As our context vector, we randomly chose any image from all trajectories with that same context, as seen in Figure 7. During testing time, we randomly generated shapes from 3, 6, and 11 units. The L2 threshold distance for success was thus the total L2 distance for all units divided by the number of units.

E. Additional Results and Hyperparameters

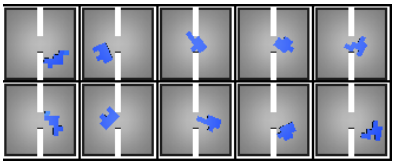

Figure 7. Example of observations (top) and contexts (bottom) of block insertion domain.

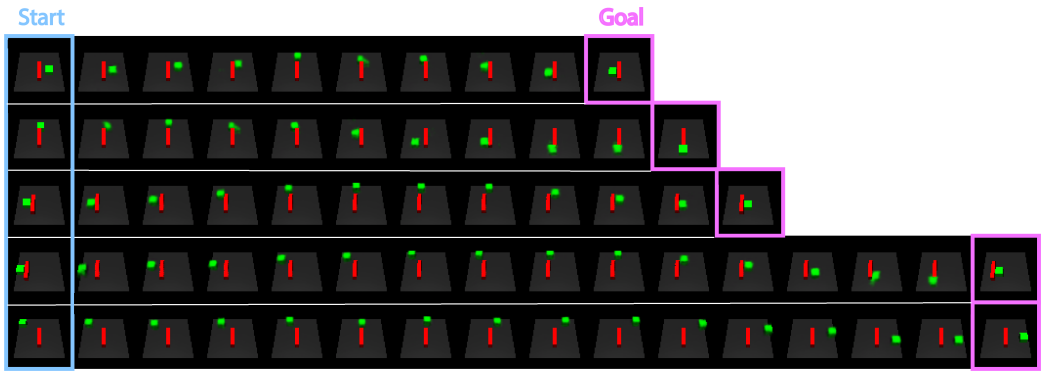

Figure 8. HTM plan examples on the block wall domain. The hallucination allows the planner to imagine how to go around the wall even though it has not seen the context before.

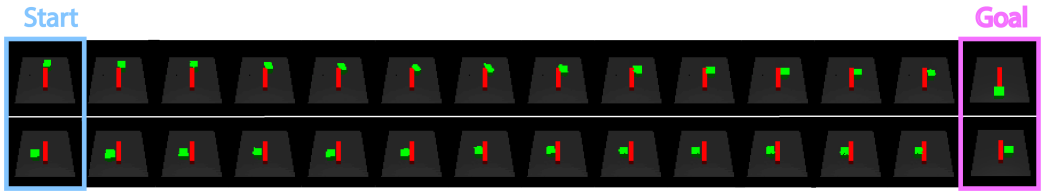

Figure 9. Visual Foresight plan examples on the block wall domain. The plans do not completely show the trajectory to the goal.

Table 3. Data parameters.

	Domain 1	Domain 2	Domain 3	Domain 4
no. contexts	150	400	360	1
initializations per context	50	30	20	1000
trajectory length	20	100	50	50
action space	$[-.05, .05]^2$	$[-.1, .1]^2$	$[-.05, .05]^4$	$[-1, 1]^2$
table size	2.8x2.8	2.8x2.8	.8x.8	.9x.7

Table 4. Planning hyperparameters.

	Domain 1	Domain 2	Domain 3
no. of samples from CVAE	300	500	300
L2 threshold for success (for each unit)	.5	.75	.1
n (timesteps to get to goal)	500	400	400
r (timesteps until replanning)	200	80	80