Entity Abstraction in Visual Model-Based Reinforcement Learning

Rishi Veerapaneni*[1] John D. Co-Reyes*[1] Michael Chang*[1]
Michael Janner[1] Chelsea Finn[2] Jiajun Wu[3] Josh Tenenbaum[3] Sergey Levine[1]

Data and Paper

Overview

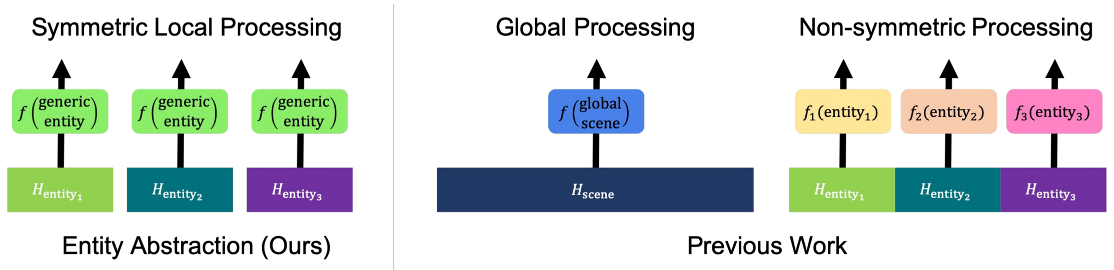
Motivation
Generalize to novel physical manipulation tasks with *compositional* structure

Idea 1: Entity Abstraction
Symmetric local processing of entities, rather than global processing of scenes, enables knowledge about an entity in one context to directly transfer to modeling the same entity in different contexts.

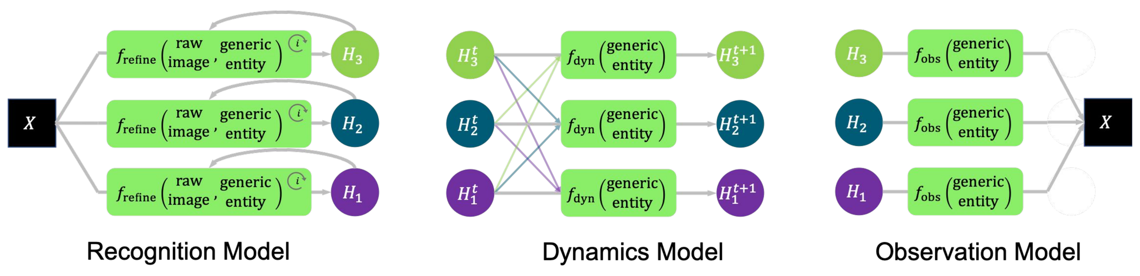
Idea 2: Interactive Inference
To ground symbolic entity variables in actual objects, use iterative inference in a entity-factorized dynamic latent variable model.

Contribution
A factorized model-based reinforcement learning framework with entity variables inferred directly from visual observations and actions without any object-level supervision.

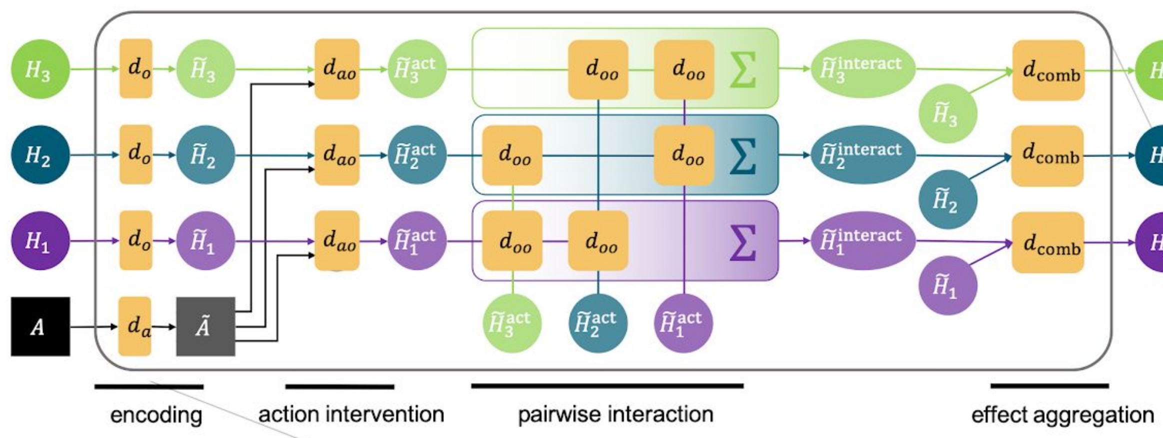
Entity Abstraction



Entity Abstraction in Modeling a POMDP

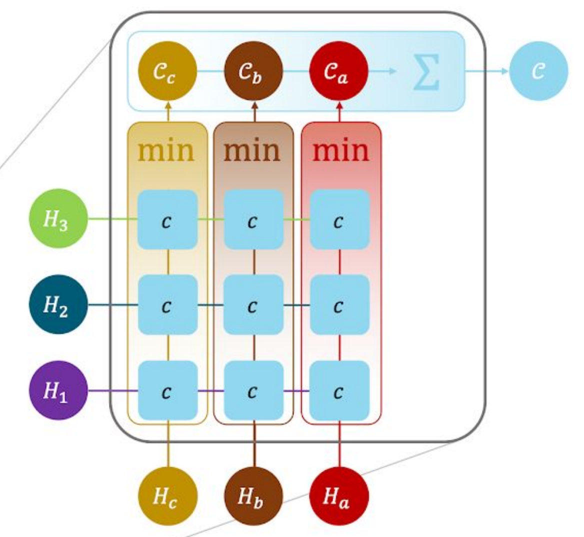


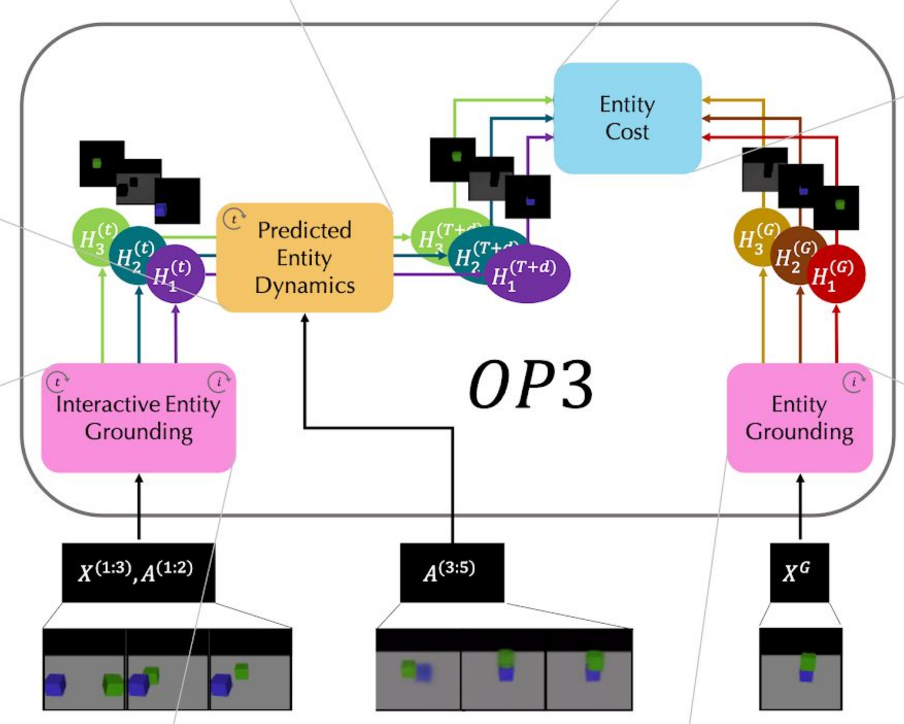
OP3: Object-centric Perception, Prediction, & Planning



Entity-Factorized Dynamics Model
The dynamics model independently evolves each entity state forward in time with the same functions by considering inertial and pairwise interactions between entities.

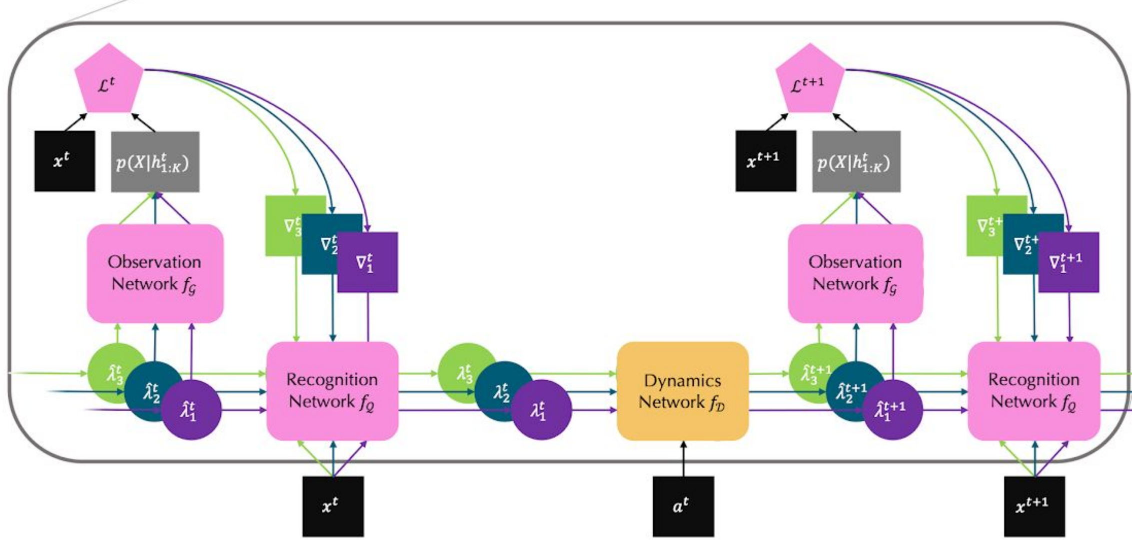
Entity-Factorized Cost Function
Modeling a scene in terms of entities enables a finer-grained way of specifying cost functions. Our cost matches each inferred goal entity latent with the closest predicted entity latent from a rollout and aggregates the cost across all latents.



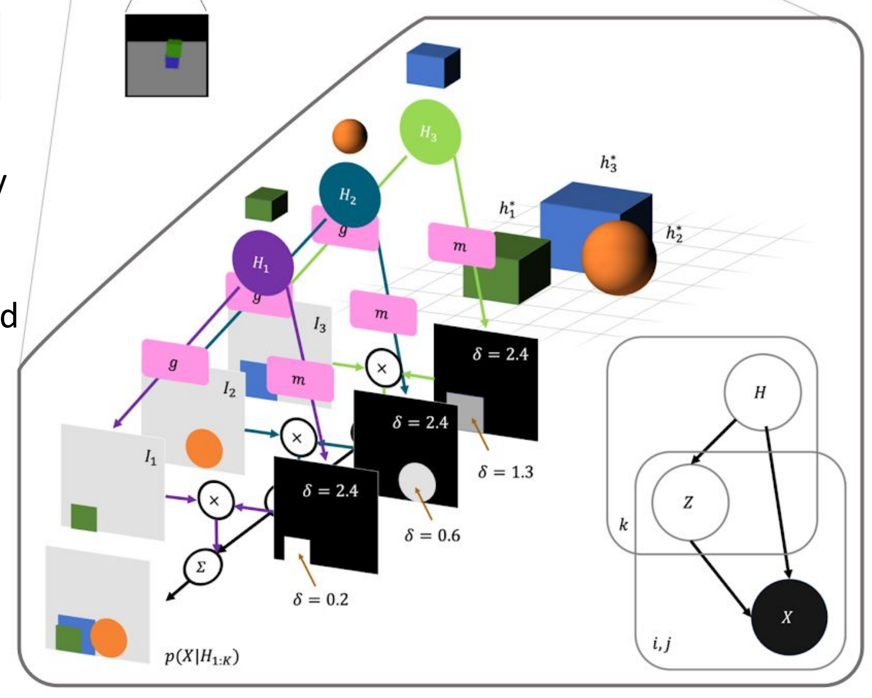


Interactive Inference for Binding Object Properties to Latent Variables
Grounding the symbolic entity variables in the actual objects of the world is a *variable binding problem*. We cast the variable binding problem as an inference problem in a dynamic latent variable model by framing the symbolic entities as random variables. OP3 infers and refines the posterior of these entities over time with neural networks.

Entity-Factorized Observation Model
The observation model is a spatial mixture model, which can be interpreted as the composition of sub-images for every entity, weighted by their depth-values which act as segmentation masks.



OP3
OP3 infers a set of latent entity states from an observation, propagates them forward in time, and scores these predicted states against inferred goal latent entity states.



Interactive Inference

Generative Distribution

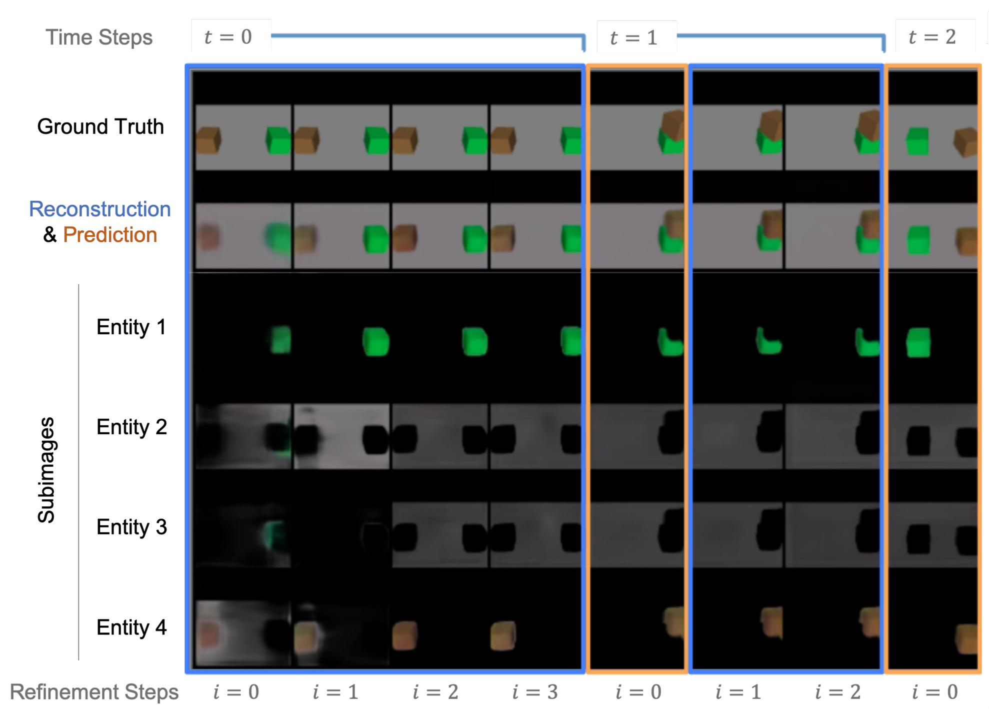
$$p\left(X^{(0:T)}, H_{1:K}^{(0:T)} \mid a^{(0:T-1)}\right) = p\left(H_{1:K}^{(0)}\right) \prod_{t=1}^{T} p\left(H_{1:K}^{(t)} \mid H_{1:K}^{(t-1)}, a^{(t-1)}\right) \prod_{t=0}^{T} p\left(X^{(t)} \mid H_{1:K}^{(t)}\right)$$

Inference Algorithm

Algorithm Interactive Inference: Timestep t

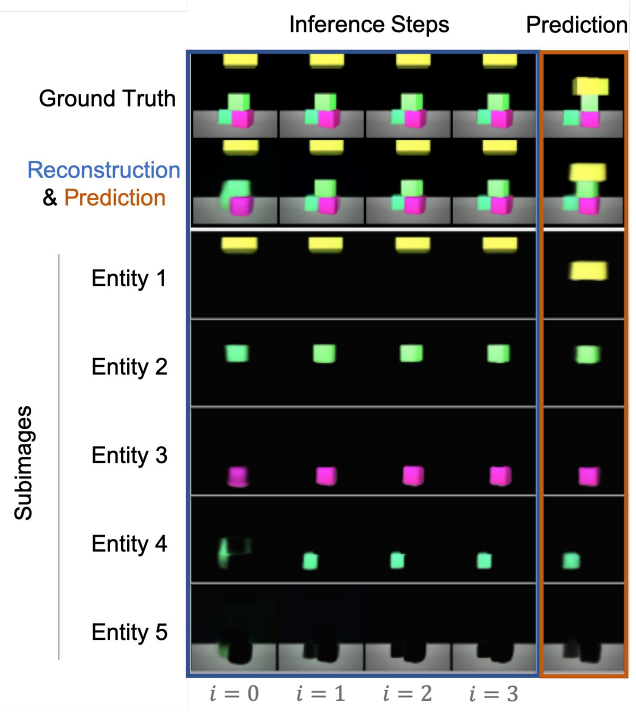
Input: observation $x^{(t)}$, action $a^{(t)}$, previous entity states $h_{1:K}^{(t-1)}$
Predict $\lambda_k^{(t,0)} \leftarrow f_d\left(h_k^{(t-1)}, h_{[\neq k]}^{(t-1)}, a^{(t)}\right)$ for each entity k
for $i = 0$ **to** $M-1$ **do**
  Sample $h_k^{(t,i)} \sim \mathcal{N}\left(\lambda^{(t,i)}\right)$ for each entity k
  Evaluate $\mathcal{L}^{(t,i)} \approx \log \mathcal{G}\left(x^{(t)} \mid h_{1:K}^{(t)}\right) - D_{KL}\left(\mathcal{N}\left(\lambda_{1:K}^{t,(i)}\right) \| \mathcal{N}\left(\lambda_{1:K}^{(t,0)}\right)\right)$
  Calculate $\nabla_{\lambda_k}\mathcal{L}^{(t,i)}$ for each entity k
  Assemble auxiliary inputs β_k for each entity k
  Update $\lambda_k^{(t,i+1)} \leftarrow f_q\left(x^{(t)}, \nabla_{\lambda_k}\mathcal{L}^{(t,i)}, \lambda_k^{(t,i)}, \beta_k^{(t,i)}\right)$ for each entity k
end for
return $\lambda^{(t,M)}$

Interactive Inference



OP3 predicts the full appearance of the green block at timestep 2 even when it was *partially occluded* before.

Single-step Rollout



OP3 isolates the prediction of the yellow block from the predictions of the others.

Planning Algorithm

Algorithm OBJECT-CENTRIC-PLANNING

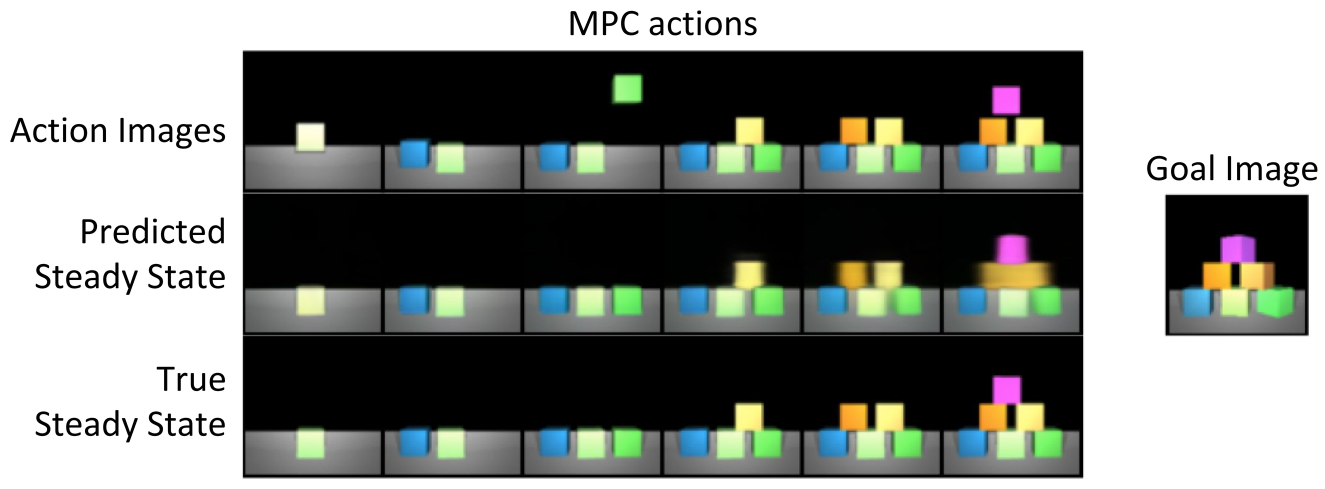
1: **Input:** $x^{(0:\tau)}, a^{(0:\tau-1)}, x^{(G)}$
2: **Initialize:** $\lambda_{1:K}^{(0)}$
3: $h_{1:K}^{(G)} \leftarrow$ GOAL INFERENCE$(\lambda_{1:K}^{(0)}, x^{(G)})$
4: $\lambda_{1:K}^{(\tau)}, h_{1:K}^{(\tau)} \leftarrow$ STATE ACQUISITION$(\lambda_{1:K}^{(0)}, x^{(0:\tau)}, a^{(0:\tau-1)})$
5: **for** $t \leftarrow \tau$ **to** T **do**
6:   $a^t \leftarrow$ ACTION SELECTION$(h_{1:K}^{(t)}, h_{1:K}^{(G)})$
7:   $x^{t+1} \leftarrow$ ENVIRONMENT-STEP(a^t)
8:   $\lambda_{1:K}^{(t+1)}, h_{1:K}^{(t+1)} \leftarrow$ STATE ACQUISITION$(\lambda_{1:K}^{(t)}, a^{(t)}, x^{(t+1)})$
9: **end for**
10: **return** $a^{(\tau:T)}$

Goal Inference: Iterative inference on the static goal image to infer goal latent entities
State Acquisition: Interactive inference on a sequence of observed images and actions to estimate the latent entities
Action Selection: Run CEM by rolling out many proposed actions sequences and scoring them with the cost function

Experiments

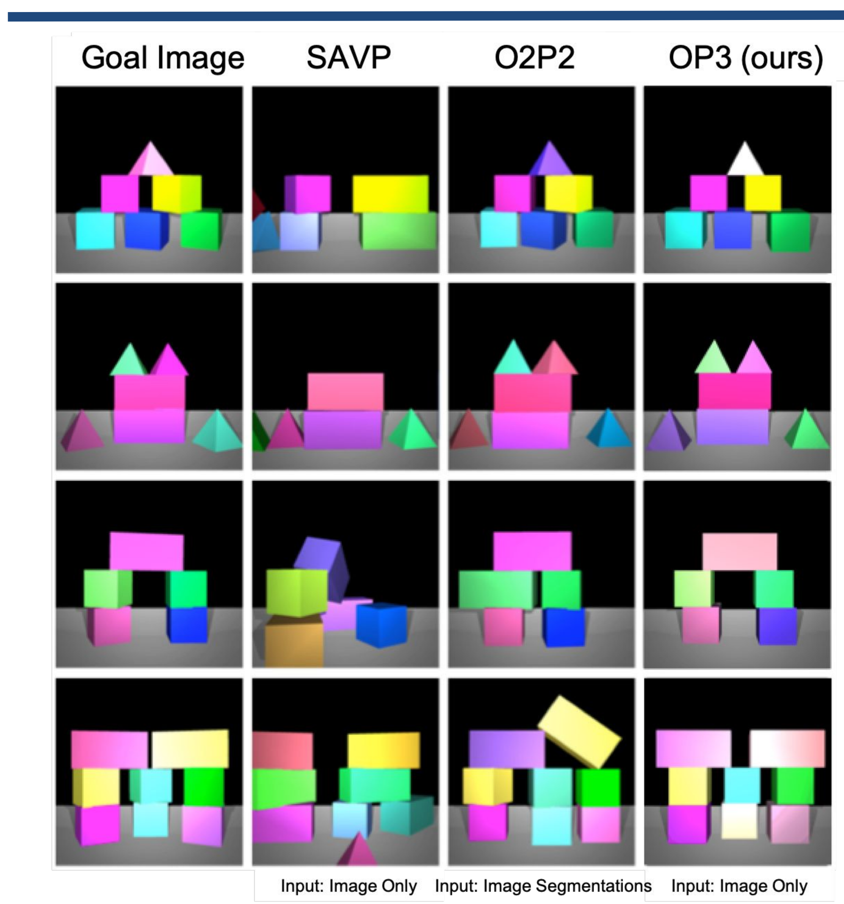
Single-Step Block-Stacking (Predict Fall)

OP3 can generalize to building complex block structures by planning in object-space from raw pixel input. We plan by first inferring the hidden states of each proposed action image, predicting the resultant steady state using our dynamics model, and then comparing the hidden states with the goal states using our cost function.



An *action image* depicts how an action intervenes on the state by raising a block in the air. OP3 is trained to predict the steady-state outcome of dropping the block.

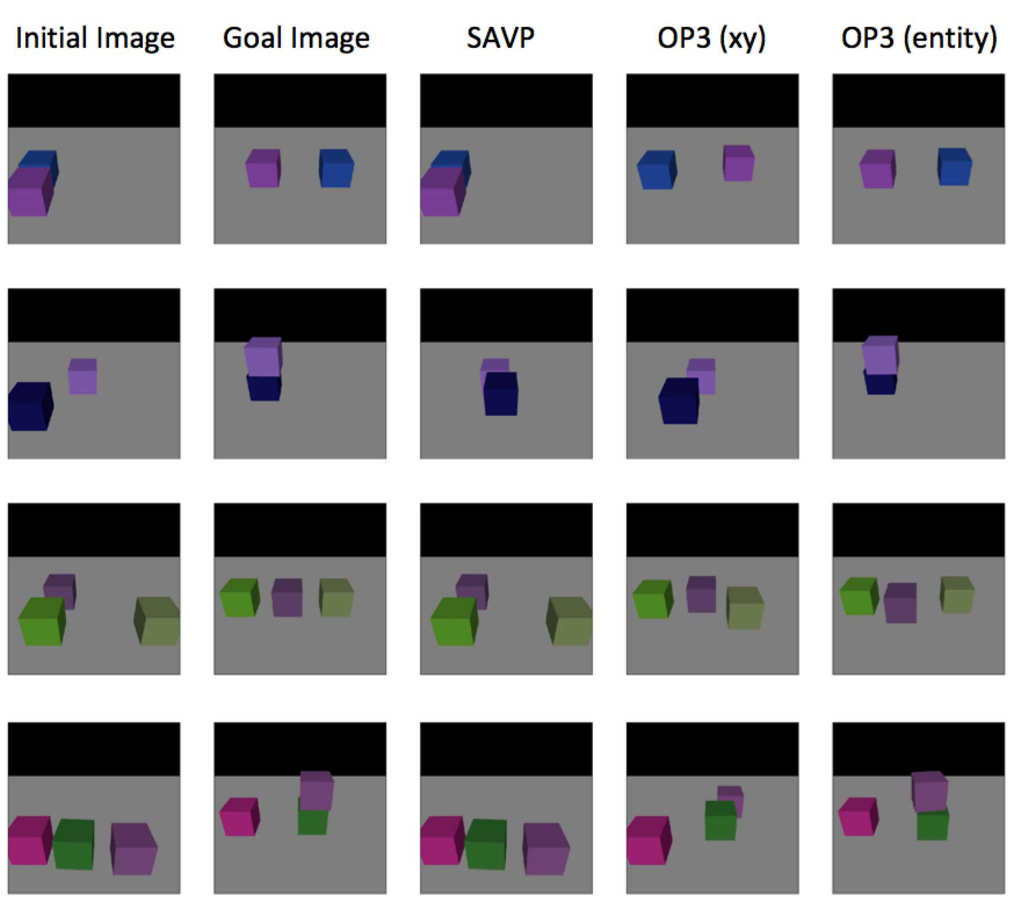
SAVP (baseline, no factorization)	O2P2 (oracle, pre-segmented input)	**OP3 (ours) (raw image input)**
24% accuracy	76% accuracy	**82% accuracy**



Results
OP3 was only trained on up to five objects but can generalize to nine objects and new configurations during testing. OP3 achieves three times the accuracy of a state-of-the-art video prediction model without entity abstraction (SAVP) and outperforms an oracle model (O2P2) that receives the object segmentations.

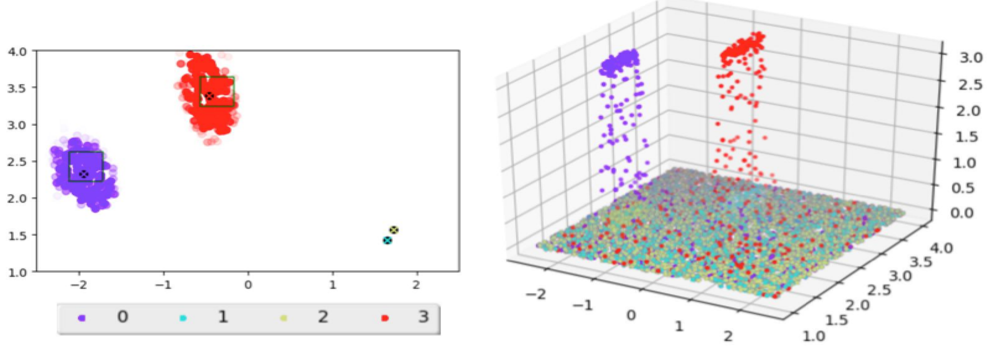
Multi-Step Block-Stacking (Pick/Place)

On a sparse block-stacking task, OP3 can plan over the space of objects multiple steps into the future.



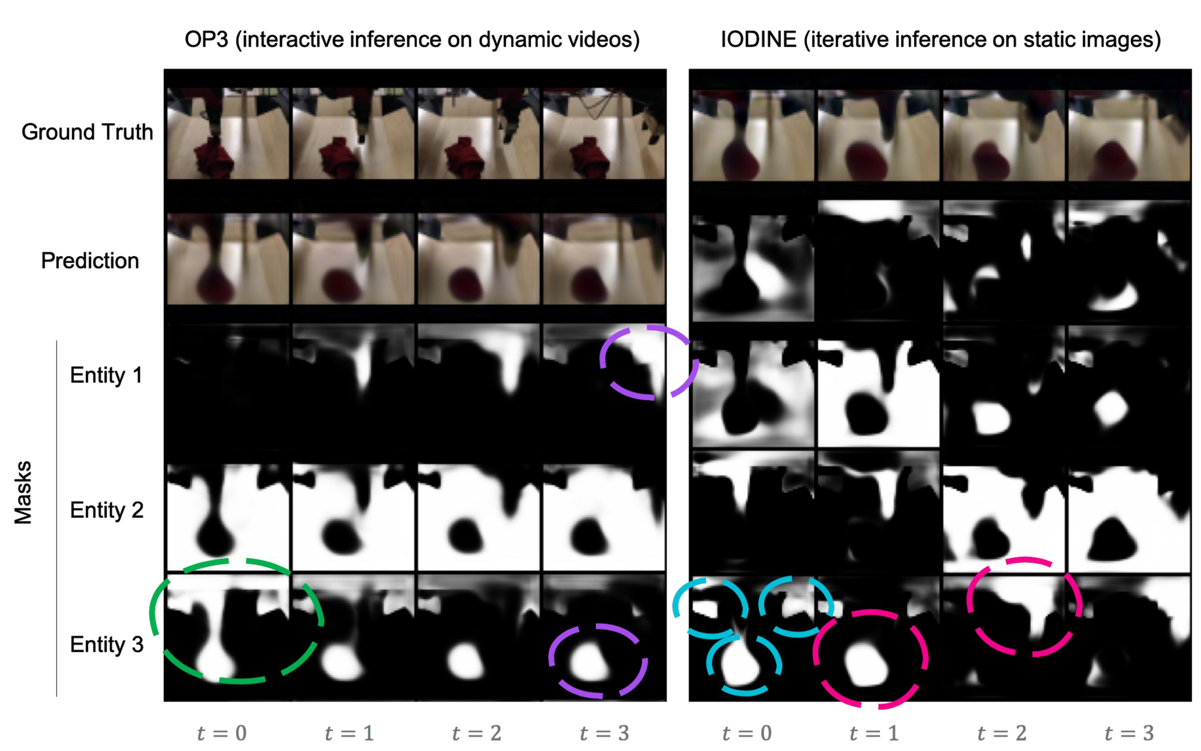
#Blocks	SAVP	OP3 (xy)	OP3 (entity)
1	54%	73%	**91%**
2	28%	55%	**80%**
3	28%	41%	**55%**

OP3 (xy): action space is `(pick_xy, place_xy)`
OP3 (entity): OP3 has access to *pointers* to each entity, enabling an *entity-centric action space* `(entity_id, place_xy)`



Real World Evaluation

We evaluate how well OP3 can disambiguate objects on a robotic pushing task with clutter and occlusions.



Initially, both OP3 (green circle) and IODINE (cyan circles) both disambiguate objects via color segmentation. As time progresses, OP3 uses temporal continuity and interactive feedback to disambiguate latents (purple), whereas applying IODINE on a per-frame basis cannot do so.

Conclusion

- OP3 integrates graphical models, symbolic computation, and neural networks in a model-based reinforcement learner
- Models as compositions of *locally-scoped* functions
- *Symbolic grounding* as *variable binding* as *posterior inference*
- Modeling entities and their interactions provides significant generalization improvement in *combinatorially complex* tasks

Challenge	Solution
Combinatorial Generalization	Entity Abstraction
Grounding Entity Variables	Interactive Inference