DeepMind

OAT: Unsupervised Object-Based Transition Models for 3D Partially Observable Environments

Antonia Creswell (tonicreswell@deepmind.com)
Rishabh Kabra, Chris Burgess, Murray Shanahan

Abstract

Existing object-based transition models **fail** to take account of **object persistence** and **object identity.** This makes it difficult to compute **semantically meaningful losses** between object representations and to integrate information about each object (and its interactions) over time.

Our contributions:

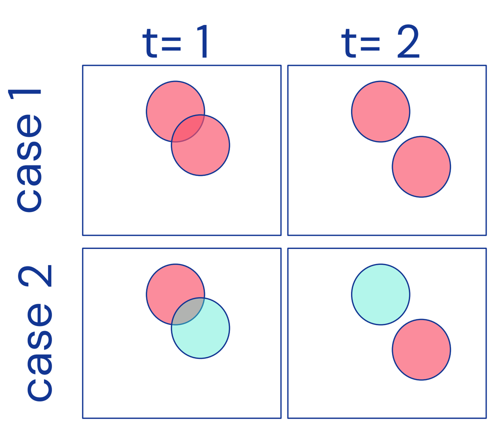
Propose an object-based transition model that:

(1) Extracts object representations.
(2) Keeps track of them over time through **alignment**.
(3) Predicts where those objects will be many steps into the future.

By dealing correctly with object persistence and identity our model is able to:

1. **Integrate information about each individual object over time leading to better long term rollouts.**
2. **Compute semantically meaningful, object-level losses to train the transition model.**

Are these balls moving towards each other or away?



For case 1, we do not know.

It is easy to compute velocity of objects when you know how they correspond between time-steps.

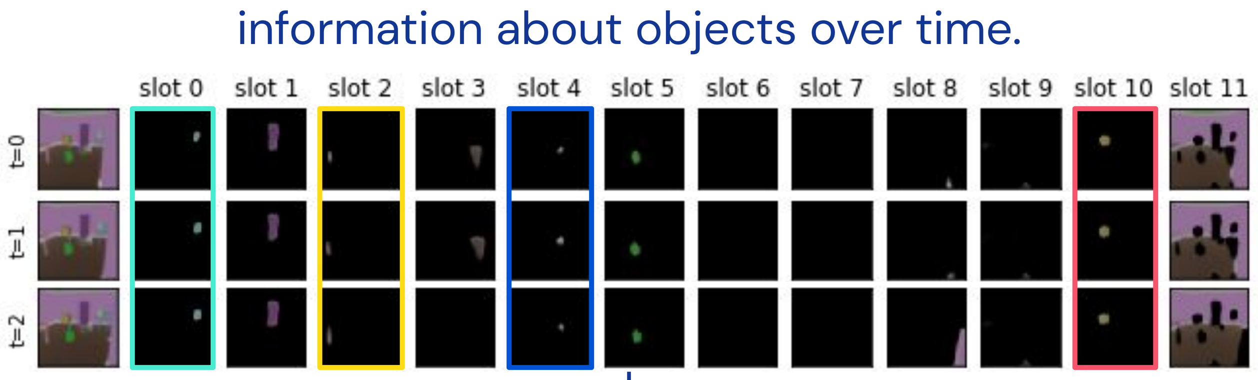
OAT outputs.

AlignNet takes unaligned MONet outputs and puts each object into a consistent slot across time.

MONet output (t=1:k)
Objects switch slots over time.

AlignNet

AlignNet output (t=1:k)
Objects are now in consistent slots across time. Making it easy to compute losses and integrate information about objects over time.

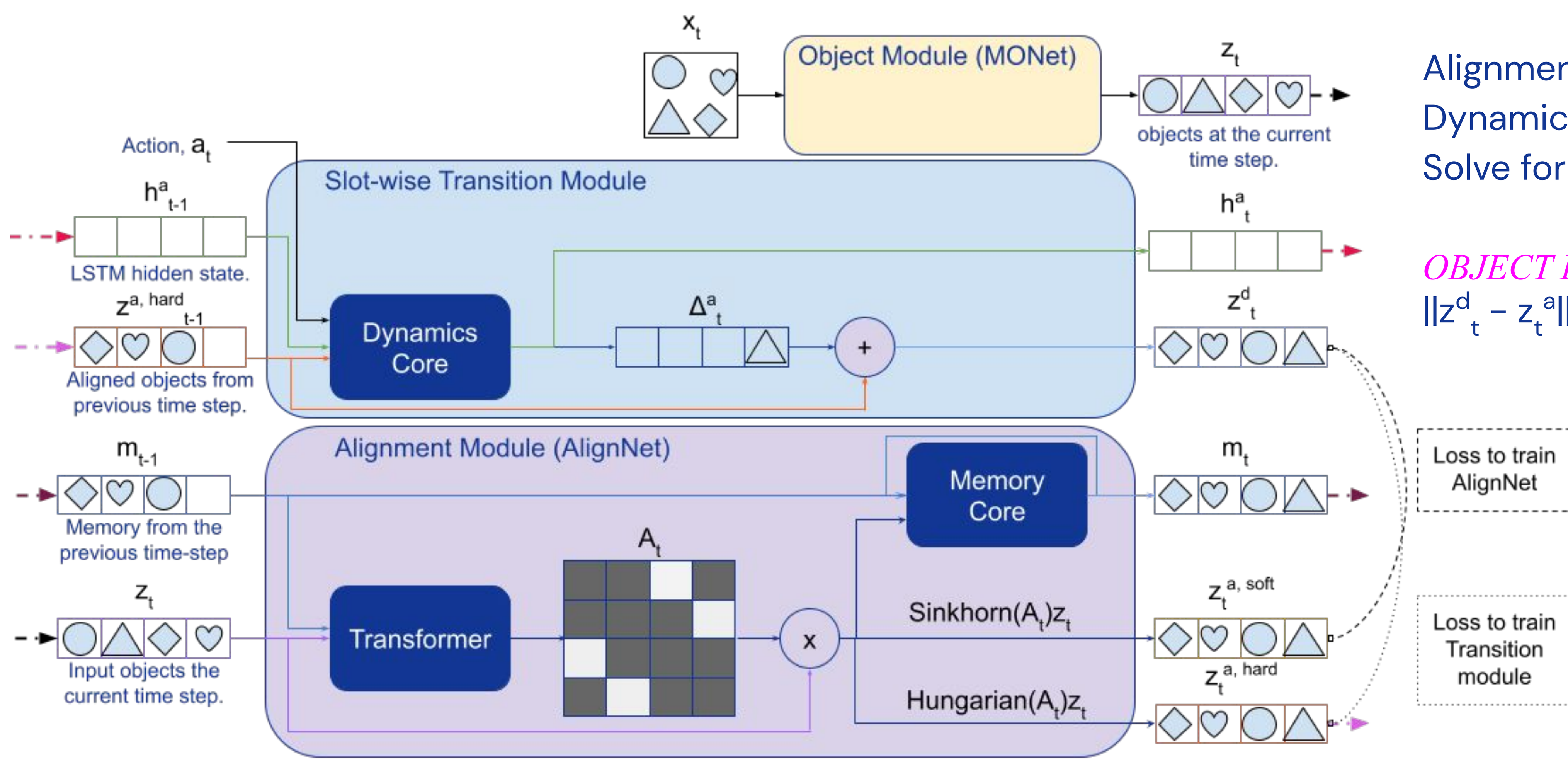


SlotWise Transition Model

Rollouts (t=k+1:T)
Object-wise predictions are in consistent slots across time.

Objects-Align-Transition

Object module (MONet): Decomposes a scene into objects and their representations.
Alignment module (AlignNet): Aligns object w.r.t. a **memory** so that objects exist in **persistent slots** across time.
Slot-wise transition module: Predicts the representation of each object at the next time-step and can be applied recurrently to produce rollouts.

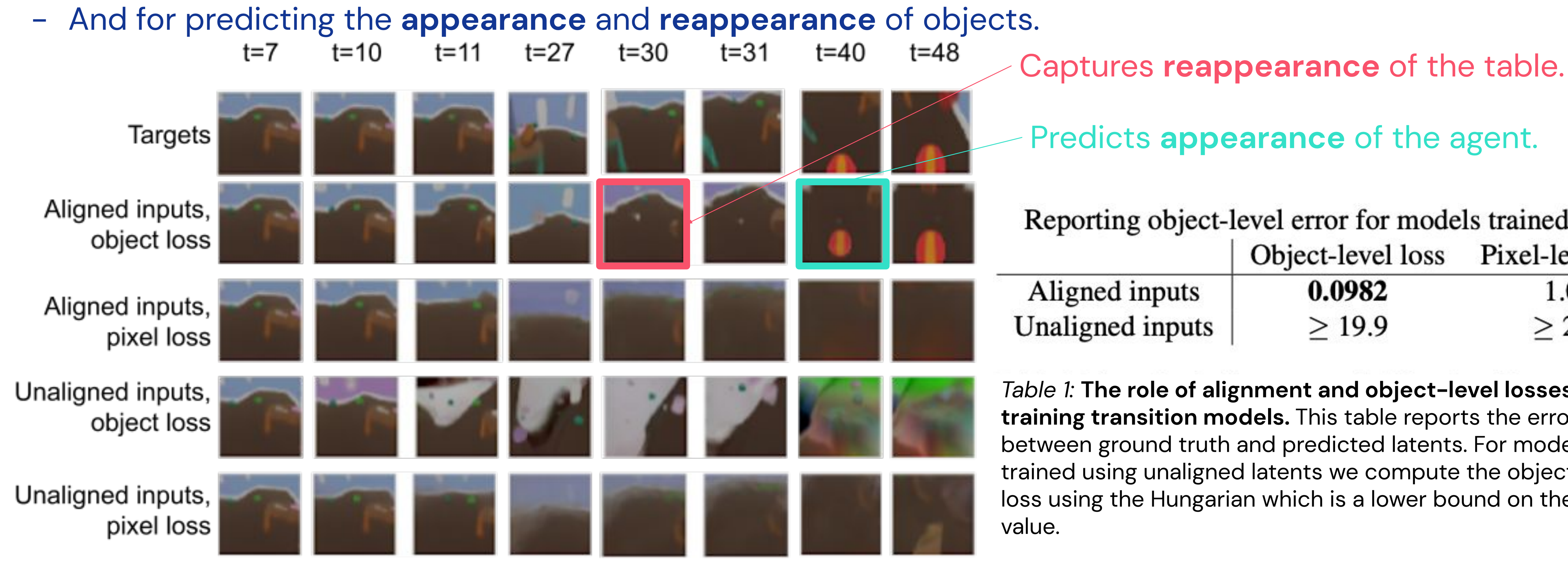


Alignment: $z^a_t = A_t z_t$
Dynamics: $z^d_t = z^a_{t-1} + \Delta_{t-1}$
Solve for $z^a_t = z^d_t$

OBJECT LEVEL-LOSS:
$||z^d_t - z_t^a|| = ||(z^a_{t-1} + \Delta_{t-1}) - A_t z_t||$

Role of Alignment when training transition models.

- Alignment and object-level loss are essential for training transition models that unroll far into the future.
- And for predicting the **appearance** and **reappearance** of objects.



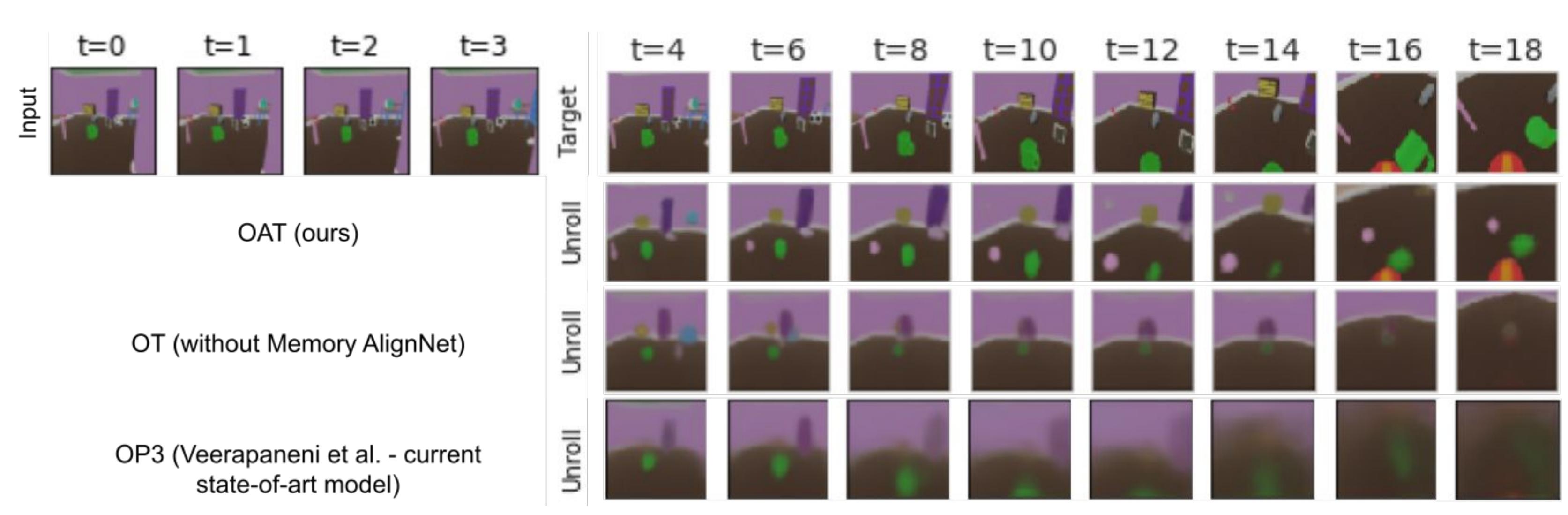
Reporting object-level error for models trained using:

	Object-level loss	Pixel-level loss
Aligned inputs	**0.0982**	1.04
Unaligned inputs	$\geq$ 19.9	$\geq$ 229

Table 1: **The role of alignment and object-level losses when training transition models.** This table reports the error between ground truth and predicted latents. For models trained using unaligned latents we compute the object-level loss using the Hungarian which is a lower bound on the actual value.

Rollouts in a 3D Playroom Environment.

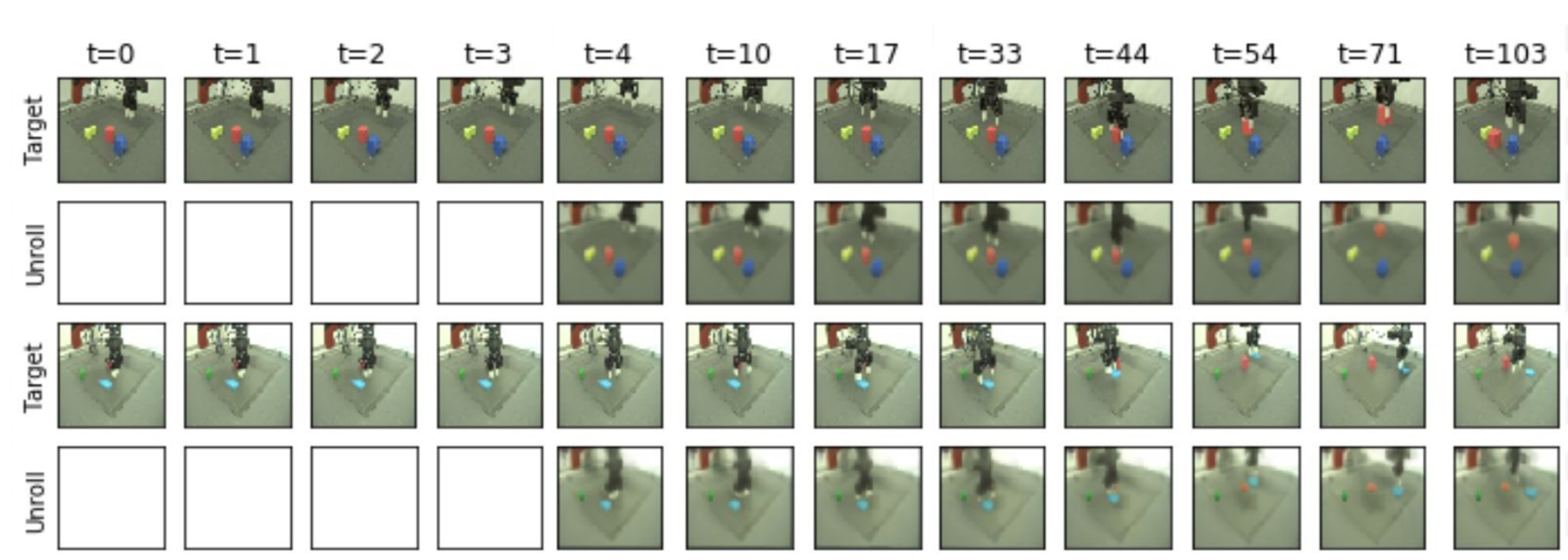
- The model is trained to unroll for 6 time-steps, here we show the model urolling for 15 time-steps.
- The model **generalises** well, unrolling for many more steps than those seen during training.



	Encoding ARI	Unroll Pixel Error	Unroll ARI
OAT (ours)	**0.62**	**0.0121**	**0.42**
OT (ours, no AlignNet)	0.59	0.0143	0.12
OP3	0.32	0.0132	0.33

Table 2: **Comparing our model OAT to baselines.** The ARI score measures how well the predicted object masks match the ground truth object masks. In the unroll we predict object representations but these can be decoded into images and masks using MONet's decoder; this is **not** used for training.

Rollouts on a Real World Robotics dataset.



- OAT performs well on this Robotics dataset (Cabi et al., 2019).
- OAT deals well with **reappearance** of the red object in the second example.