

Table 3: The Architecture and Hyperparameters of Encoder Learning.

Name	Value						
	bigfish	caveflyer	coinrun	dodgeball	jumper	ninja	starpilot
# slots	31	63	31	31	31	47	47
slot dimension	256						
# slot attention iterations	3						
image size	[224, 224, 3]						
patch size	16						
optimizer	Adam						
learning rate	4e-4						
data size	1M						
epoch	15						
batch size	64						
Cosmos model	Cosmos-0.1-Tokenizer-CI16x16						

Table 4: The Architecture and Hyperparameters of Dynamics Learning.

Name	Value
dynamic feature dimension	256
static feature dimension	256
self-attention # layers	1
self-attention model size	512
self-attention # heads	8
SSM # layers	2
SSM model size	512
SSM d_{state}	64
SSM d_{conv}	4
optimizer	Adam
learning rate	1e-4
batch size	32

A Method Details

A.1 Encoder

The architecture and hyperparameter used during encoder training are shown in Table. 3.

A.2 Dynamics

During dynamic feature training, the slot reconstruction loss and the disentanglement loss may have conflicting gradient and hinder model learning. To mitigate this issue, we leverage gradient modification [24] to enhance the balance between two objectives.

The architecture and hyperparameter used during encoder training are shown in Table. 4.

B Experiment Details

B.1 Evaluating Object-Centric Representation

The object-centric representation evaluation for 4 procgen environments are shown in Table. 5 and Fig. 7. By leveraging segmentation masks only during training, Dyn-O significantly outperforms SOLV in all environments, in terms of slot-object binding accuracy.

B.2 Evaluating World Model Accuracy

The world model accuracy for other procgen environments are shown in Table. 6 and Fig. 8 - Fig. 10. In most environments, Dyn-O significantly outperforms baselines in term of prediction accuracy.

Table 5: slot-object binding accuracy, measured by FR-ARI (\uparrow).

Environments	Oracle	Dyn-O (ours)	SOLV
bigfish	0.96	0.80	0.54
coinrun	0.33	0.27	0.10
dodgeball	0.79	0.48	0.17
starpilot	0.86	0.47	0.49
average	0.74	0.51	0.33

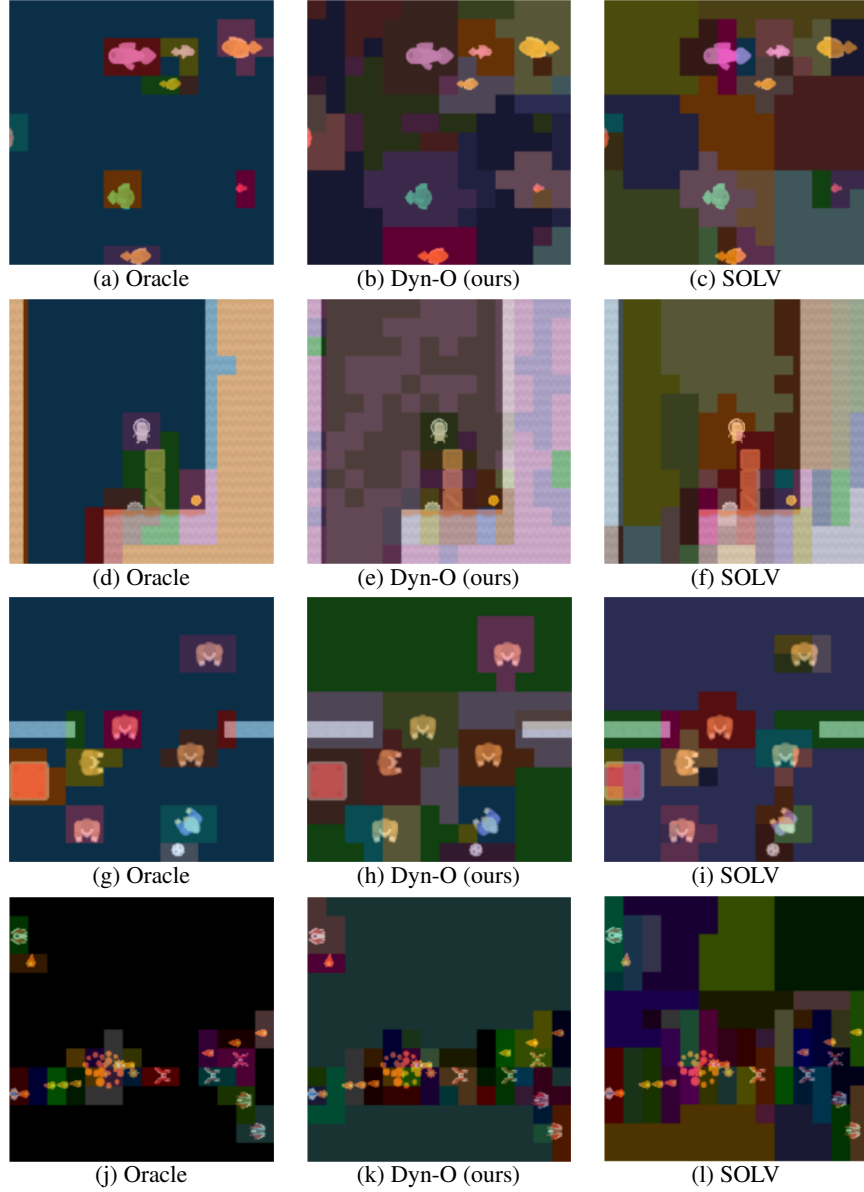


Figure 7: Qualitative evaluation of the object-centric representation learning in **bigfish**, **coinrun**, **dodgeball**, and **starpilot**.

410 B.3 Evaluating Static-Dynamic Disentanglement

411 The probing accuracy for other progen environments are shown in Table. 7 - 10. The same privilege
 412 information may belong to different properties in different environments, which we label out in the

Table 6: Rollout accuracy at 20-th timestamp, measured as mean and standard error.

Environment	Metric	DreamerV3	Dyn-O w/o OC	Dyn-O (ours)
bigfish	LPIPS (\downarrow)	0.39 ± 0.01	0.36 ± 0.01	0.25 ± 0.01
	FVD (\downarrow)	567.20 ± 11.18	248.94 ± 16.32	126.88 ± 6.42
	SSIM (\uparrow)	0.73 ± 0.01	0.68 ± 0.01	0.76 ± 0.01
	PSNR (\uparrow)	19.34 ± 0.14	20.16 ± 0.27	20.28 ± 0.30
caveflyer	LPIPS (\downarrow)	0.53 ± 0.01	0.59 ± 0.01	0.61 ± 0.01
	FVD (\downarrow)	1104.50 ± 18.87	966.07 ± 28.71	877.74 ± 20.99
	SSIM (\uparrow)	0.44 ± 0.01	0.26 ± 0.01	0.26 ± 0.01
	PSNR (\uparrow)	11.99 ± 0.12	10.93 ± 0.19	10.82 ± 0.13
coinrun	LPIPS (\downarrow)	0.34 ± 0.01	0.36 ± 0.01	0.28 ± 0.01
	FVD (\downarrow)	530.31 ± 10.51	583.11 ± 16.07	266.13 ± 6.16
	SSIM (\uparrow)	0.60 ± 0.01	0.47 ± 0.01	0.62 ± 0.01
	PSNR (\uparrow)	14.22 ± 0.25	12.61 ± 0.14	13.56 ± 0.19
dodgeball	LPIPS (\downarrow)	0.42 ± 0.01	0.15 ± 0.01	0.14 ± 0.01
	FVD (\downarrow)	903.51 ± 20.44	481.55 ± 20.83	372.50 ± 15.07
	SSIM (\uparrow)	0.50 ± 0.01	0.72 ± 0.01	0.76 ± 0.01
	PSNR (\uparrow)	16.10 ± 0.05	20.33 ± 0.12	20.88 ± 0.15
ninja	LPIPS (\downarrow)	0.48 ± 0.01	0.49 ± 0.01	0.45 ± 0.01
	FVD (\downarrow)	423.27 ± 14.26	521.32 ± 15.85	313.57 ± 9.73
	SSIM (\uparrow)	0.45 ± 0.01	0.33 ± 0.01	0.54 ± 0.01
	PSNR (\uparrow)	12.58 ± 0.17	12.80 ± 0.10	11.95 ± 0.12
starpilot	LPIPS (\downarrow)	0.37 ± 0.01	0.50 ± 0.01	0.22 ± 0.01
	FVD (\downarrow)	626.47 ± 14.95	429.36 ± 15.46	211.27 ± 12.70
	SSIM (\uparrow)	0.69 ± 0.01	0.72 ± 0.01	0.76 ± 0.01
	PSNR (\uparrow)	19.99 ± 0.08	19.76 ± 0.23	20.55 ± 0.13

Table 7: Probing accuracy (\uparrow), in percentage (%), on **bigfish** privilege properties.

	mean R values static	mean G values static	mean B values static	x position dynamic	y position static	area static
slots	74.7 ± 0.0	77.9 ± 0.0	83.8 ± 0.0	97.7 ± 0.0	98.3 ± 0.0	100.0 ± 0.0
dynamic features	49.3 ± 3.0	48.4 ± 2.4	47.4 ± 3.7	94.0 ± 1.9	45.2 ± 5.1	95.9 ± 1.2
static features	65.8 ± 4.2	67.6 ± 4.8	77.5 ± 1.2	29.3 ± 0.4	88.8 ± 0.7	100.0 ± 0.0
random features	37.6 ± 0.0	31.8 ± 0.0	37.3 ± 0.0	19.2 ± 0.0	20.8 ± 0.0	88.6 ± 0.0

413 tables. In some environments, the same privilege information can be static properties of some objects
414 and can dynamic for other objects. In such case, we mark such privilege information as "mixed".

Table 8: Probing accuracy (\uparrow), in percentage (%), on **dodgeball** privilege properties.

	mean R values static	mean G values static	mean B values static	x position mixed	y position mixed	area static
slots	90.2 ± 0.0	91.7 ± 0.0	91.2 ± 0.0	98.7 ± 0.0	98.7 ± 0.0	99.8 ± 0.0
dynamic features	66.0 ± 1.6	62.4 ± 1.2	61.1 ± 1.8	55.4 ± 1.8	57.7 ± 2.5	95.3 ± 1.7
static features	73.1 ± 1.2	74.2 ± 1.4	75.5 ± 1.7	74.9 ± 2.4	70.4 ± 2.9	99.3 ± 0.0
random features	53.3 ± 0.0	35.5 ± 0.0	42.9 ± 0.0	20.1 ± 0.0	19.7 ± 0.0	90.1 ± 0.0

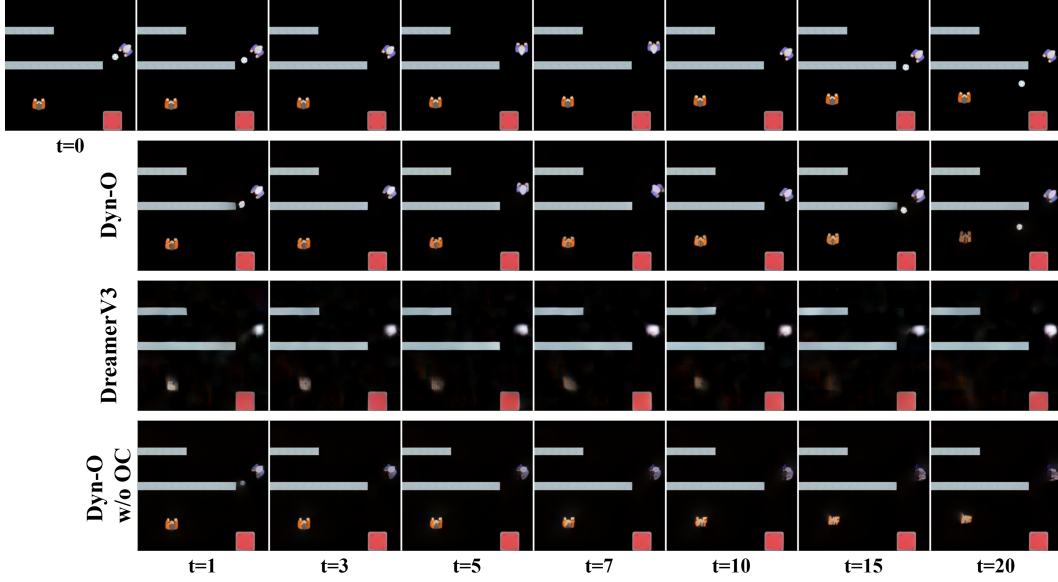


Figure 8: 20-step rollouts in **dodgeball**. **1st** row: ground-truth, **2nd** row: Dyn-O (ours), **3rd** row: DreamerV3, and **4th** row: Dyn-O w/o OC. Dyn-O significantly outperforms dreamer, with sharp player shape and accurate predictions of threw balls.

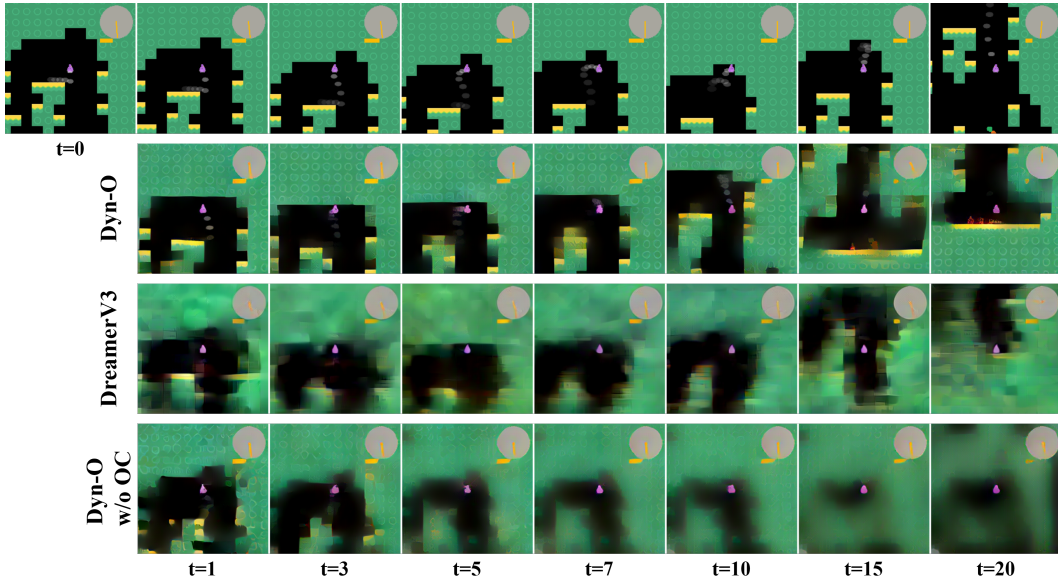


Figure 9: 20-step rollouts in **jumper**. **1st** row: ground-truth, **2nd** row: Dyn-O (ours), **3rd** row: DreamerV3, and **4th** row: Dyn-O w/o OC. Dyn-O significantly outperforms dreamer, with sharp wall and player trail until 10th timestamp.

Table 9: Probing accuracy (\uparrow), in percentage (%), on **ninja** privilege properties.

	mean R values static	mean G values static	mean B values static	x position dynamic	y position dynamic	area dynamic
slots	87.1 ± 0.0	80.7 ± 0.0	86.1 ± 0.0	95.0 ± 0.0	96.6 ± 0.0	97.3 ± 0.0
dynamic features	60.7 ± 9.7	51.3 ± 8.2	60.6 ± 9.0	86.3 ± 3.0	88.1 ± 0.3	87.6 ± 2.2
static features	77.6 ± 2.9	62.2 ± 0.4	79.6 ± 0.4	35.9 ± 0.9	37.8 ± 1.1	82.2 ± 1.3
random features	32.9 ± 0.0	25.4 ± 0.0	31.3 ± 0.0	25.0 ± 0.0	19.5 ± 0.0	80.5 ± 0.0

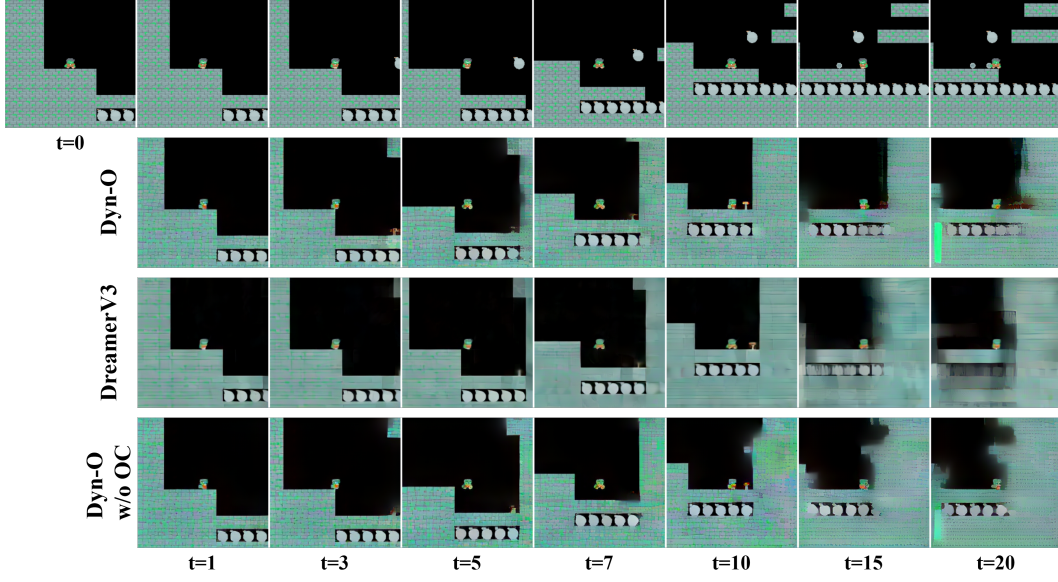


Figure 10: 20-step rollouts in **ninja**. **1st** row: ground-truth, **2nd** row: Dyn-O (ours), **3rd** row: DreamerV3, and **4th** row: Dyn-O w/o OC. Dyn-O significantly outperforms dreamer, with sharper wall shape at 15-th timestamp.

Table 10: Probing accuracy (\uparrow), in percentage (%), on **starpilot** privilege properties.

	mean R values static	mean G values static	mean B values static	x position dynamic	y position static	area static
slots	71.5 ± 0.0	79.1 ± 0.0	71.0 ± 0.0	97.2 ± 0.0	97.5 ± 0.0	99.5 ± 0.0
dynamic features	55.6 ± 7.2	66.5 ± 6.4	55.3 ± 6.2	94.8 ± 0.9	77.0 ± 14.2	97.9 ± 1.7
static features	55.9 ± 1.1	70.7 ± 1.1	50.5 ± 1.8	26.8 ± 0.7	76.6 ± 3.2	99.2 ± 0.0
random features	35.9 ± 0.0	50.6 ± 0.0	36.3 ± 0.0	18.3 ± 0.0	22.7 ± 0.0	92.0 ± 0.0