COMB: A FACTORIZED OBJECT-CENTRIC WORLD MODEL AND ITS VALUE FOR POLICY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Object-centric world models decompose scenes into entities, but each entity is typically represented by a single unstructured embedding. We study whether policy learning also benefits from a second decomposition, in which each entity is represented by named attributes such as position and appearance. This factorization can, in principle, support transfer to attribute combinations that were never observed during training. We introduce COMB, an object-centric world model with explicitly factorized per-entity latents, learned without attribute supervision and trained for online model-based RL in imagination. It achieves the highest or tied-highest return on eight 2D and 3D manipulation tasks and leads the baselines on held-out attribute combinations. To separate representation format from information content, we freeze COMB and insert an invertible map between its representation and the policy heads. The map preserves all information, changes only the coordinates in which it is expressed, and can be varied continuously. On 2D and 3D tasks with held-out position–colour combinations, entangling the attributes reduces return by 31–37% at a fixed budget, with larger losses at higher doses, and reduces held-out return by up to 30% at matched in-distribution skill. Linear scrambling produces no resolvable loss. These results suggest that, once entities are separated, policy learning benefits from making task-relevant attributes linearly readable; the effect therefore depends on both the representation and the head that reads it.

1 INTRODUCTION

World models learn environment dynamics from experience and use the learned model to train an agent in imagination (Ha & Schmidhuber, 2018; Hafner et al., 2025a). Object-centric world models (OCWMs) go a step further by representing a scene as a set of entities (Greff et al., 2020). This structure has been linked to better compositional generalization and data efficiency: object-centric representations can support robust property prediction, be less sensitive to some out-of-distribution (OOD) shifts (Dittadi et al., 2022), and generalize compositionally better than dense representations when data or compute are limited (Kapl et al., 2025).

It is much less clear whether these benefits translate to policy learning. Object-centric agents have reported gains on relational manipulation and visually complex scenes (Mosbach et al., 2025; Haramati et al., 2024; Zhang et al., 2025), but a systematic study of object-centric pre-training for RL found improvements mainly on relational-reasoning tasks, with similar or worse performance elsewhere and no consistent advantage out of distribution (Yoon et al., 2023). Likewise, a world model with disentangled object-level latents can reconstruct and predict well while training weaker policies than DreamerV3 (Ferraro et al., 2025b); under distribution shift, a scene-centric world model using similar pretrained features can remain competitive with object-centric alternatives (Nazirjonov et al., 2026). These comparisons change several ingredients at once, including the encoder, the information it retains, the head that reads the representation, and the coordinates in which that information is expressed. As a result, they do not isolate which property of an object-centric representation, if any, benefits a policy.

We isolate one such property. An OCWM separates a scene into entities, but each entity is still usually an unstructured embedding. We instead consider a second decomposition in which each entity is written as a sum of per-attribute codes, such as position and appearance. This structure

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

should help a policy because, with an additive code, a policy that has seen a red object on the left, a red one on the right, and a blue one on the right can already read the corresponding blue-left state (Section 3), without observing every combination. This is a claim about representation *format*, not *content*: an invertible map applied after a frozen encoder changes the coordinates while preserving all information, so format can be tested with content held fixed. Our contributions are:

- **COMB**, an object-centric world model whose per-entity latents are factorized into position, scale, depth, and appearance without attribute supervision, trained for online model-based RL in imagination and evaluated against holistic and slot-based world models on eight 2D and 3D tasks (Sections 4–5).
- **Generalization Arena**, a manipulation benchmark built around held-out attribute combinations, distractors, and relational variants (Section 5).
- **Entanglement dosing**, an intervention that scrambles a frozen representation by a controlled amount while preserving its information exactly. With a calibrated placebo and conditioning-matched linear controls, it turns the usual binary ablation into a dose–response experiment (Sections 6–7).

2 Related Work

World models and object-centric world models. Latent world models learn dynamics from pixels and train behaviour in imagination or by planning. The Dreamer family (Hafner et al., 2019; 2021; 2025a) and TD-MPC2 (Hansen et al., 2024) are strong baselines for sample-efficient online control, alongside transformer-based and large-scale video world models (Micheli et al., 2023; Zhang et al., 2023; Bruce et al., 2024; Hafner et al., 2025b). Their state is a single holistic vector with no explicit entity or attribute boundaries, so transfer across object configurations must be learned by the downstream heads. Object-centric world models instead represent a scene as a set of slot embeddings (Burgess et al., 2019; Greff et al., 2019; Locatello et al., 2020), extended to video (Kipf et al., 2022; Wu et al., 2023) and to real-world scenes through pretrained features (Seitzer et al., 2023; Zadaianchuk et al., 2023), and supporting relational prediction and planning (Veerapaneni et al., 2020; Kossen et al., 2020). Recent work reports gains in online RL on relational and visually complex tasks (Ferraro et al., 2025a; Mosbach et al., 2025; Zhang et al., 2025; Nishimoto & Matsubara, 2025; Feng et al., 2025). Each slot, however, is still an unstructured embedding: the scene is decomposed into entities, not into attributes.

Factorized particle representations. Another line of work represents each entity with explicit attributes. SCALOR and G-SWM separate an object's location from its appearance (Jiang et al., 2020; Lin et al., 2020), while Deep Latent Particles (Daniel & Tamar, 2022; 2024) use particles with named position, scale, depth, transparency, and appearance coordinates. These representations have been used for model-free goal-conditioned RL (Zadaianchuk et al., 2021; Haramati et al., 2024), diffusion policies learned from demonstrations (Qi et al., 2025), and LPWM (Daniel et al., 2026), a particle world model trained end-to-end on video with per-particle latent actions and conditioning on actions, language, or goal images. LPWM is the closest model to ours and uses the same 2D particle, but it learns no reward, value, or policy from interaction, and ECRL learns no world model; COMB does both, training reward, critic, and actor online in imagination. This online setting is central to our question because policy training determines which attribute combinations are actually visited. Appendix A compares these models property by property (Table 2).

3 Why Should Structure Help a Policy?

A policy can only learn from states it encounters. Its reward head, critic, and actor are fitted on the states visited during training and then applied to states that may be new. The relevant question is therefore how well what the heads learn on visited states transfers to states they have not seen. Even a simple example shows that this depends on how the representation is written.

A toy example. Consider an object with a *position* attribute p (left or right) and an *appearance* attribute f (red or blue). Suppose training has seen a red object on the left, a red one on the right, and a blue one on the right, but never a blue one on the left. A *factorized* representation writes the

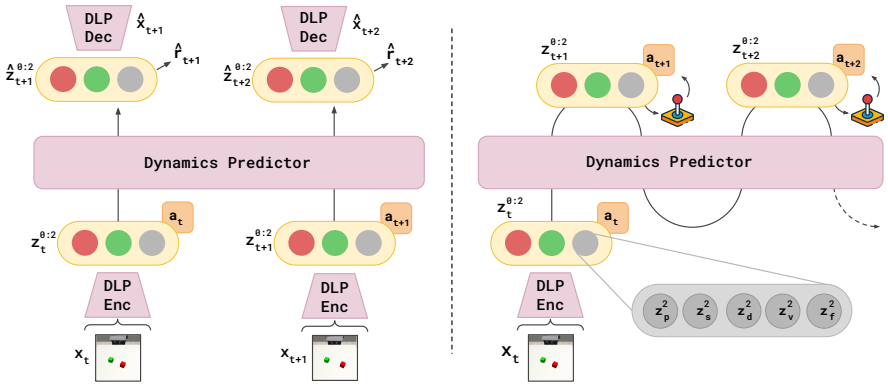


Figure 1: Architecture of COMB. Particles carry named position and appearance coordinates; the aggregator is the only module that mixes across entities, and the reward gradient is stopped at its input.

object as one code per attribute, added together,

$$z(p, f) \;=\; e_p + c_f. \tag{1}$$

With this code, the unseen combination is determined by the three seen ones,

$$z(\text{left}, \text{blue}) \;=\; z(\text{left}, \text{red}) + z(\text{right}, \text{blue}) - z(\text{right}, \text{red}), \tag{2}$$

and any linear readout $w^\top z$ satisfies the same identity. Two linear readouts that agree on the three seen combinations must therefore agree on the fourth: the learned mapping extends to blue-left even though that combination was never observed. More generally, with P position values and F appearance values, $P{+}F{-}1$ well-chosen combinations determine all $P{\times}F$ combinations. When two attributes share one part of the factorized representation, as shape and colour do in COMB's appearance code, there is no equivalent shortcut. Still, the effect of an unseen appearance is localized to that code: if the new code differs by ϵ, a linear readout changes by $w^\top \epsilon$ and nothing else. An entangled code provides no analogous bound.

4 COMB

We build on Dynamic Deep Latent Particles (DDLP) (Daniel & Tamar, 2022; 2024), a VAE-based method for unsupervised object-centric representation learning that decomposes an observation $\boldsymbol{x}_t$ into K low-dimensional latent "particles" $\{\boldsymbol{z}_t^k\}_{k=1}^K$ ($k{=}0$ is a learned background slot), evolved by a dynamics predictor and rendered back to pixels by a decoder (Appendix C.1). What matters here is how a particle is *written*:

$$\boldsymbol{z}_t^k \;=\; \big(\underbrace{\boldsymbol{z}_{p,t}^k}_{\text{position}},\, \boldsymbol{z}_{d,t}^k,\, \boldsymbol{z}_{s,t}^k,\, \boldsymbol{z}_{v,t}^k,\, \underbrace{\boldsymbol{z}_{f,t}^k}_{\text{appearance}}\big), \qquad \begin{matrix} \boldsymbol{z}_{p,t}^k, \boldsymbol{z}_{s,t}^k \in \mathbb{R}^2, & \boldsymbol{z}_{d,t}^k \in \mathbb{R}, \\ \boldsymbol{z}_{v,t}^k \in [0,1], & \boldsymbol{z}_{f,t}^k \in \mathbb{R}^F \end{matrix} \tag{3}$$

This structure comes from architectural inductive biases rather than attribute supervision. A spatial-softmax bottleneck localizes each particle at position $\boldsymbol{z}_{p,t}^k$; a spatial transformer uses that position and the scale $\boldsymbol{z}_{s,t}^k$ to crop a glimpse, whose convolutional encoding becomes the appearance code $\boldsymbol{z}_{f,t}^k$. The decoder then places the glimpses back on a canvas using depth $\boldsymbol{z}_{d,t}^k$ and visibility $\boldsymbol{z}_{v,t}^k$ to determine their layering. Equation 3 therefore has the factorized form used in Section 3: position and appearance occupy separate coordinates, and those coordinates have explicit meanings.

To build a world model we condition the dynamics predictor on the agent's actions, $\boldsymbol{h}_{t+1}^{0:K} = f_\phi(\boldsymbol{z}_{0:t}^{0:K}, \boldsymbol{a}_{0:t})$, so that dynamics operates in a structured latent space where information about each entity stays separated across particles. Policy learning and reward prediction require relational reasoning *across* entities, so a learned *particle aggregator* g_ξ (Figure 1) maps $\boldsymbol{z}_t^{0:K}$ to a fixed-size

summary $\boldsymbol{g}_t$ used by all downstream heads. Following DreamerV3 (Hafner et al., 2025a), the reward head outputs a categorical distribution over a symlog-spaced two-hot target,

$$\mathcal{L}_{\text{reward}}(\xi) \;=\; -\,\mathbb{E}_t\big[\log r_\xi\big(r_t \,\big|\, g_\xi\big(\text{sg}\big[\boldsymbol{z}_t^{0:K}\big]\big)\big)\big]\,, \tag{4}$$

where $\text{sg}[\cdot]$ denotes a stop-gradient. The reward is relational: for example, it can depend on the distance between an object and its goal. If this gradient were allowed to shape the encoder, the easiest solution could be to encode such relations directly into a particle. That would let one token represent multiple entities and weaken the one-particle-per-entity structure on which the factorization depends. With the stop-gradient, the encoder is instead shaped by reconstruction and dynamics, while relations are computed in the aggregator (Appendix C.4).

Aggregation and behaviour learning. The aggregator g_ξ is the only module in the policy and value pathway that mixes information across particles. We concatenate each particle's attributes into a token $\boldsymbol{p}_t^k$ and process the resulting set with a two-stage transformer adapted from the Entity Interaction Transformer (EIT) of Haramati et al. (2024). The first stage uses permutation-equivariant self-attention; the second uses cross-attention with learned queries to pool the set into $\boldsymbol{g}_t$. The reward head, actor, and critic each use an *independent* aggregator (Appendix C.2). The aggregator begins with a trainable linear projection W_{proj}. As shown in Section 6, this projection can absorb any linear scramble, while Section 7 measures the part that it cannot. We train a DreamerV3-style actor–critic on $\boldsymbol{g}_t$ for $H{=}10$ imagination steps and use an online MPPI planner, following TD-MPC2 (Hansen et al., 2024), to select actions. Details are in Appendix C.5.

5 COMB: In- and Out-of-Distribution Policy Learning

Setup. We evaluate three task families spanning 2D and 3D, sparse and dense rewards, and discrete and continuous actions (insets of Figure 2; Appendix B): the sparse-reward 2D **OCRL** suite (Yoon et al., 2023) (*Reach*, *OddOneOut*, and *OOO-Hard* over 5 shapes $\times$ 5 colours), the dense-reward 3D **MOF-Reach** (Mosbach et al., 2025), and our **Generalization Arena (GA)**, a Robosuite (Zhu et al., 2020) manipulation benchmark whose objects are parameterized by shape, colour, and spawn region. Baselines are **DreamerV3** (Hafner et al., 2025a), a holistic latent world model; **TD-MPC2** (Hansen et al., 2024), a model-predictive baseline without pixel reconstruction; and **SOLD** (Mosbach et al., 2025), a slot-based world model whose slots are unstructured embeddings. We use the public implementations with default hyperparameters, parameter counts matched to COMB as closely as practical, $64{\times}64$ RGB observations, and an update every 5 environment steps (Appendix C.6). All methods receive the same number of environment steps; in 3D, COMB and SOLD start policy learning after 50k steps, so the other baselines get 50k more policy-learning steps. COMB, including every dose arm, acts through an MPPI planner initialized by its actor, as in TD-MPC2, whereas DreamerV3 acts directly from its actor.

Held-out combinations. A compositional claim requires held-out *combinations* of attributes, which existing manipulation suites rarely provide. The Generalization Arena is designed around this setting. We use two types of held-out split, and the distinction matters for the analyses that follow. Most splits hold out *appearance*: shape $\times$ colour combinations in OCRL-OOO-Hard and GA-ShapeColor, an unseen target colour among distractors in GA-Distractor, or an unseen hue band in GA-OOO. COMB represents shape and colour jointly in its learned appearance code. Two splits instead hold out combinations of appearance and *position*, which is inferred by a separate mechanism and therefore provide the direct test of the factorization considered in Section 3. In GA-Cube, a red or blue cube can spawn in one of four quadrants, with 2 of the 8 (colour, quadrant) cells held out; this turns the toy example into a manipulation task. OCRL-Reach uses 4 colours and 4 quadrants and holds out 4 of 16 cells. Exact splits and their motivation are given in Appendices B.1 and B.2.

In distribution. COMB has the highest mean final return on seven of the eight tasks and ties TD-MPC2 on MOF-Reach (Figure 2). The difference is most pronounced on OCRL, where identifying the task-relevant objects is an important exploration bottleneck. COMB also reconstructs and predicts more accurately than DreamerV3 (Appendix G.2), an advantage separate from how its code is organized.

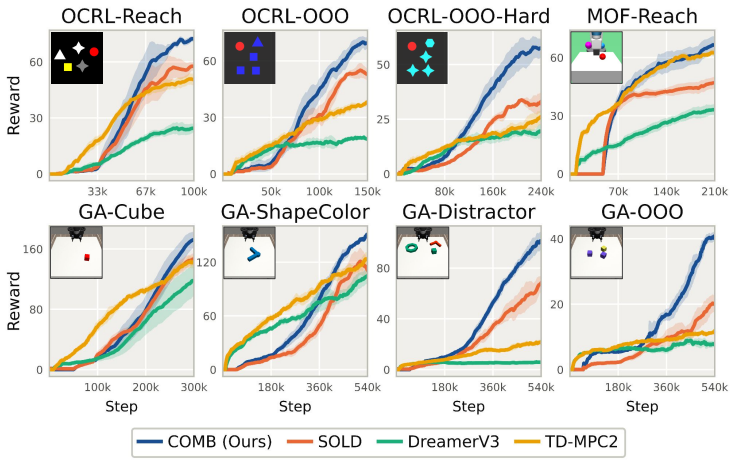


Figure 2: Online policy learning curves across eight control tasks, each shown by an inset render (full resolution in Figure 4). Shaded regions denote standard error over three seeds.

OOD evaluation	COMB	SOLD	DreamerV3	TD-MPC2
appearance × *appearance*				
OCRL-OOO-Hard 6/25	**56.4** ± 4.3 (−2%)	24.2 ± 3.1 (−25%)	18.6 ± 2.3 (−3%)	30.2 ± 3.7 (+18%)
GA-ShapeColor 2/9	**121.1** ± 5.2 (−18%)	90.8 ± 7.6 (−21%)	81.0 ± 9.9 (−21%)	41.2 ± 4.3 (−66%)
GA-Distractor 1/4 col.	**100.9** ± 7.1 (0%)	64.5 ± 4.7 (−3%)	4.3 ± 1.7 (−26%)	9.3 ± 2.1 (−56%)
GA-OOO hue band	**36.9** ± 3.2 (−8%)	14.6 ± 4.1 (−27%)	4.5 ± 1.5 (−42%)	9.2 ± 1.9 (−19%)
position × *appearance*				
OCRL-Reach 4/16	**63.0** ± 1.9 (−12%)	50.8 ± 3.0 (−11%)	28.9 ± 3.8 (+20%)	39.9 ± 3.7 (−21%)
GA-Cube 2/8	**160.9** ± 5.4 (−6%)	83.1 ± 7.2 (−43%)	74.7 ± 12.4 (−36%)	81.5 ± 7.7 (−43%)

Table 1: Held-out return at the budgets of Figure 2 (OCRL-Reach 100K, OCRL-OOO-Hard 250K, GA-Cube 300K, other GA 550K), 3 seeds × 10 episodes. In parentheses: change relative to the method's own final in-distribution return. Next to each task: held-out cells (Appendix B). At low in-distribution return the relative change is dominated by noise (positive changes and grey entries).

Out of distribution. On held-out attribute combinations, COMB has the highest return among the compared methods, but the pattern differs by the type of shift (Table 1). When the task requires ignoring a distractor, separating entities already accounts for much of the difference: the slot-based model narrows the gap to COMB, while the holistic models remain far lower. When the target must instead be identified by comparing attributes across entities, entity separation alone helps less, and the slot-based model falls back toward the holistic baselines. The lower block holds out position × colour, the setting of Section 3. On GA-Cube, COMB loses 6% of its in-distribution return on held-out cells, compared with losses of 36–43% for the baselines, including SOLD. On OCRL-Reach the differences are smaller: COMB and SOLD lose similar shares (12% and 11%), TD-MPC2 loses 21%, and COMB retains the highest held-out return.

What this cannot show. Figure 2 and Table 1 are the usual end-to-end comparison and do not isolate the source of the gap: COMB differs from the baselines in encoder, information content, and heads, and its perception is also better. On appearance × appearance splits, the encoder may also map an unseen (shape, colour) pair onto a code seen in training (Appendix G.1). Isolating format requires holding encoder, information, and head fixed and changing only the coordinates the head receives.

6 DOSING REPRESENTATION ENTANGLEMENT

We next ask what scrambling a factorized code should cost a policy, and build an instrument that scrambles a frozen COMB by a controlled amount while preserving its information exactly.

Scrambling the code. Consider an invertible map ψ applied to the code from Section 3 before it reaches any head. The transformed code $\psi(z)$ contains exactly the same information as z, but Eq. 2 is preserved only when ψ is linear, since linear maps are the ones that preserve sums and differences.

Proposition 1 (What a head learns from seen combinations)**.** *Consider linear readouts fitted on a set $\mathcal{S}$ of seen (p, f) combinations that links every attribute value to every other through a chain of seen combinations (the smallest such set has $P+F-1$ elements).*

(a) ***A factorized code transfers.*** *Two readouts that agree on $\mathcal{S}$ agree on all $P\times F$ combinations.*

(b) ***Linear scrambling is free.*** *Part (a) still holds after any invertible linear map, however badly conditioned.*

(c) ***Nonlinear scrambling is not.*** *After an invertible nonlinear map, part (a) can fail. In the extreme, two readouts can agree on $\mathcal{S}$ yet take arbitrary values on every other combination, so all $P\times F$ must be seen.*

In the toy example, the three seen combinations form the required chain (proofs in Appendix D). Part (b) matters most in practice: our heads are nonlinear, but each begins with a trainable linear layer that absorbs any linear ψ, so whether a representation is effectively entangled depends on the head that reads it. Parts (a) and (b) are close to results on extrapolation with additive models (Lachapelle et al., 2023) and identifiability of object-centric codes (Brady et al., 2023), while disentanglement alone does not guarantee compositional generalization through learned decoders (Montero et al., 2021; Schott et al., 2022); our contribution is to test this distinction inside a policy-learning system.

Measuring the damage. Real transformations lie between the two extremes of Proposition 1. Since only the nonlinear part of ψ matters, we fit the best affine map from original to scrambled tokens and record the share of variance it cannot explain, the *entanglement dose*

$$\eta(\psi) \;=\; 1 - R^2\big(\text{best affine fit to } \psi\big) \;\in\; [0,1], \tag{5}$$

with $R^2{=}1-\sum \mathrm{SSE}/\sum \mathrm{SST}$ pooled over all coordinates of the tokens as the heads receive them, on held-out episodes. It is 0 when ψ is linear on the data and grows as more of ψ escapes the first linear layer. With bounded readout weights, disagreement on an unseen combination is at most proportional to the nonlinear residual of ψ (Appendix D); the bound is not a function of η, so a cost rising with dose is an empirical prediction.

What this predicts. In RL, "seen" means visited, so Proposition 1 applies to ordinary training as well as to held-out tests. It predicts:

- **P1. A linear scramble costs nothing**, however badly conditioned (part (b)).
- **P2. A nonlinear scramble lowers return at a fixed budget, increasingly with η**, because each visited combination tells the heads less about those not yet visited (part (c) and the graded bound).
- **P3. Out of distribution, it erodes transfer.** At matched in-distribution skill, a dose-zero policy should act on unseen combinations of attributes held in separate codes as well as on seen ones (part (a)), and a dosed policy should fall short, increasingly with η. A more expressive head should not close the gap, since capacity cannot supply a constraint the data never contained.

Construction. We implement this continuum with a map ψ_λ applied to the tokens of a *frozen* COMB at the input of every head. Each token $\boldsymbol{p}$ is split into halves $(\boldsymbol{u}, \boldsymbol{v})$; each coupling layer permutes coordinates, so that across layers every coordinate is updated equally often and position and appearance coordinates mix, and then sets

$$\boldsymbol{u} \leftarrow \boldsymbol{u}, \qquad \boldsymbol{v} \leftarrow \boldsymbol{v} + s_i\, t_i(\boldsymbol{u}), \tag{6}$$

with t_i a frozen random MLP and s_i calibrated so that $\mathrm{std}(s_i t_i(\boldsymbol{u})) = \lambda\,\mathrm{std}(\boldsymbol{v})$. Each layer inverts in closed form, so information preservation is a *guarantee* rather than a measurement, and its Jacobian is unit triangular ($\det J{=}1$), so the map neither compresses nor expands the token distribution. A frozen COMB with a given mixer is an *arm*, and arms of increasing λ form a *ladder*; λ is only a handle, and we report the dose η each arm realizes on its own suite.

Two controls a switch cannot express. All arms share the same whitening, clamp, and calibration pipeline (Appendix E.1). The *calibrated placebo*, a frozen random orthogonal map inside this pipeline, controls for the apparatus itself. The *conditioning-matched linear twins* address the concern that nonlinear maps are merely harder to optimize: each is a linear map matched to a coupling arm's conditioning, one per axis because no linear map can match both Jacobian and covariance conditioning, so a coupling arm that underperforms its twin is not explained by conditioning on that axis. With the identity, these form the *dose-zero controls*.

7 What Entanglement Costs a Policy

7.1 Offline: Controlled Coverage

We first test Proposition 1 without exploration, optimization, or imagination: on frozen COMB tokens we bin position and appearance into a $4{\times}4$ grid, build an additive target, hold out 4 cells while keeping the chain condition, and fit a linear head and an MLP on the rest (Appendix D, Table 6). The identity and placebo recover the held-out cells exactly (parts (a) and (b); P1). With the linear head, held-out error rises with η and exceeds the error on seen cells at every nonzero dose (part (c)). The MLP fits the seen cells nearly perfectly at every dose yet still loses on held-out cells, so capacity does not recover what the scramble removed (P3).

Mixing within attributes. Our mixer crosses position and appearance coordinates, which leaves open whether the cost requires crossing attributes. It does not: the same flow applied within the position and within the appearance coordinates separately, keeping each attribute in its own coordinates, gives at matched dose ($\eta{=}0.06$ against 0.05) a held-out risk of 0.25 against 0.04 with the linear head and 0.18 against 0.02 with the MLP, and lowers the linear readability of position to $R^2{=}0.54$ against 0.98 (Table 7). Separate coordinates are therefore not enough; each attribute must also stay linearly readable. The same control shows that η, a variance share over the whole token, can understate damage confined to a few coordinates, which is why Section 7.3 measures readability directly.

7.2 Online: Inside the Full RL System

We sweep the same dial inside the full RL system. Two suites are the position–colour splits of Section 5, OCRL-Reach (2D, sparse reward) and GA-Cube (3D, dense reward), the clearest test of P3 because COMB infers position and colour by separate mechanisms; the third, OCRL-OOO-Hard, crosses shape with colour and defines the target by comparing appearance across objects. For each suite we freeze a pretrained COMB, shared by all arms, and train reward head, actor, and critic from scratch on the mixed tokens of the training cells only (on OCRL-Reach, the minimal chain of 7 of 16 cells rather than the 12 of Table 1); since only the heads are retrained, these returns are not comparable with Table 1. Budgets, seeds, and per-arm results are in Appendix E.2.

The cost grows with the dose. Across all suites, in-distribution return at a fixed budget decreases as η increases (Figure 3b). At the highest dose, the policy loses 37% of COMB's return on OCRL-Reach, 31% on GA-Cube, 60% on OCRL-OOO-Hard, and 52% on MOF-Reach, the fourth suite without held-out cells (Appendix E.2). Along each ladder all arms train on the same cells for the same budget and read the same information, only in different coordinates.

A linear scramble costs nothing. The dose-zero arms (the identity, an orthogonal placebo, and the linear twins) remain within 6% of COMB on every suite, with no difference resolved from it (P1). This band sets the resolution of the experiment; every dosed arm on the two position $\times$ colour suites falls below it, the weakest already by 10–11%. The linear twins rule out conditioning as the cause: they stay inside the band although each is matched to a coupling arm that loses 18–28%

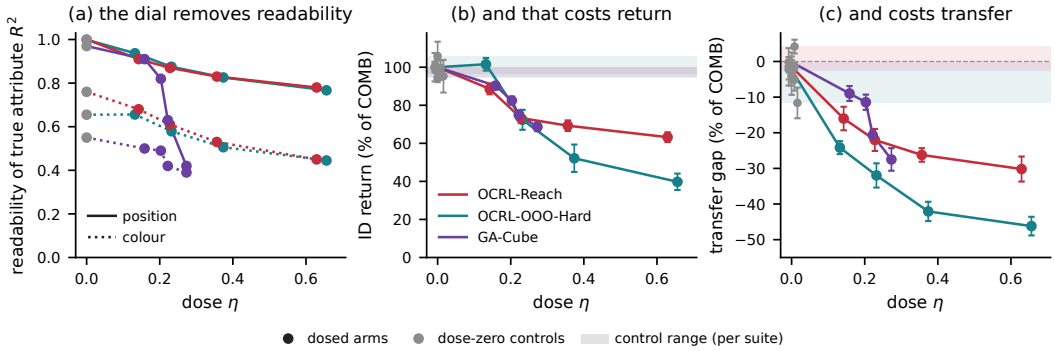


Figure 3: **One dial, three suites.** Every arm is an exact bijection of COMB's representation, and each suite is plotted against its own measured η. **(a)** R^2 of an affine probe from the representation the heads receive to each object's true position (solid) and colour (dotted). **(b)** In-distribution return at the end of a fixed budget, % of COMB. **(c)** Transfer gap: held-out minus in-distribution return at matched in-distribution skill, % of COMB. Grey points: dose-zero controls; tinted bands: each suite's control range. Mean ± s.e.m. over seeds; per-arm numbers in Table 8.

(Appendix E.2). The remaining difference is therefore the nonlinearity measured by η, and at a fixed budget the entangled representation leads to lower return from the same experience (P2). A standard binary ablation using only a linear mixer would have seen at most the 6% control range and could therefore miss this effect.

7.3 From Dose to Readability and Transfer

What the dial removes. The dose measures entanglement of the whole token, not which attributes a head can no longer read, and Section 7.1 identifies per-attribute readability as what matters. We therefore probe each arm with an affine map, held out by episode, from the representation the heads receive to each object's true position and colour; affine, because the aggregator begins with a per-token linear projection. COMB and all controls read position at $R^2 \geq 0.97$, and readability of both attributes falls with dose on every suite (Figure 3a). Since every arm is a bijection, no information is removed, only linear access to it, which the heads must rebuild from the states they visit. R^2 compares within a suite only, because in 3D the fixed camera confines the workspace to a small part of the frame (Appendix E.2). On an absolute scale, over all 18 arms of OCRL-OOO-Hard and MOF-Reach, return as a share of COMB's falls with the probe's absolute position error at $r{=}{-}0.89$ (-0.97 over the dosed arms), whereas R^2 correlates with it only at $r{=}0.56$.

Transfer to unseen combinations. Proposition 1 concerns combinations the policy never visits, so we measure transfer at matched skill, since held-out return at the end of training conflates transfer with how much was learned. The *transfer gap* is held-out minus in-distribution return, averaged over evaluations where in-distribution return lies between 60% and 90% of COMB's final return; its sign holds for every band we tested (Appendix E.2). On the position $\times$ colour suites the dose-zero policies keep their performance on held-out cells (gaps from -3% to $+4\%$ of COMB's return), even on OCRL-Reach, whose heads are trained on the minimal chain of 7 of 16 cells. This is part (a) in the full system: a held-out cell reaches the heads as a seen position code plus a seen colour code. After mixing, it becomes a point the visited cells do not constrain in the same way (part (c)), and the dosed arms lose 9–30% of COMB's return, more at higher dose ($p{<}10^{-4}$ against the controls on both suites; Figure 3c). The dial therefore costs transfer, not only learning speed (P3).

The gap follows what the task reads. On OCRL-OOO-Hard the transfer gap is larger, 24–46% against at most 12% for the controls, although the dial damages both OCRL representations similarly (Figure 3a) and OCRL-Reach uses fewer training cells. What differs is the role of the held-out attribute. On OCRL-Reach and GA-Cube colour is a nuisance: the policy acts on position, which a held-out cell leaves at a seen value. On OCRL-OOO-Hard the held-out cell changes the appearance the policy compares across objects to find the target, so distortion reaches the decision directly. The

controls' loss of up to 12% there fits the case in Section 3 of two attributes sharing one code, and the encoder may also map an unseen pair onto a seen code (Appendix G.1). Transfer can also degrade first: at η=0.13 the policy matches COMB in distribution but loses 24% on held-out combinations.

8 Discussion and Conclusion

We asked whether attribute factorization, beyond entity separation, matters for policy learning when information content is held fixed. COMB, which combines both with online model-based RL, attains the highest or tied-highest return on all eight tasks and all held-out splits at comparable parameter and sample budgets. The dose experiments cleanly isolate representation *format*: entangling coordinates while keeping underlying information intact lowers return and reduces transfer to unvisited attribute combinations, with performance penalties scaling monotonically with dose.

Each part of Proposition 1 mirrors dynamics in the full system. Part (a) establishes the transfer mechanism: factorizing attributes into separate codes enables policy heads to generalize across combinations, reducing the required environmental interactions from the product to the sum of attribute values. However, separate codes alone are insufficient if policy readouts cannot isolate them (Section 7.1); transfer requires each attribute to remain linearly readable, which COMB provides for spatial coordinates by construction. While dose experiments fix entity separation to isolate attribute formatting, entity separation itself remains key for ignoring distractors (Section 5). Part (b) clarifies representation "format": because a trainable linear layer absorbs linear scrambles, a linearly mixed code—even one classified as entangled by standard disentanglement metrics (Locatello et al., 2019)—does not impair policy learning. Disentanglement utility is thus a joint property of the representation together with the head that reads it (Higgins et al., 2017; van Steenkiste et al., 2019; Locatello et al., 2019).

Part (c) reconciles conflicting findings in prior literature. Representation costs are graded rather than binary, frequently surfacing on unvisited attribute combinations before impacting in-distribution return. Consequently, mild scrambles or in-distribution-only evaluations act as placebos, obscuring representation defects. Because held-out attribute combinations also test encoder coverage (Appendix G.1), representation ablations in RL should adhere to three guidelines: measure and report scrambling dose, evaluate held-out alongside in-distribution performance at matched skill, and test attribute combinations generated by separate encoder mechanisms.

Named coordinates move beyond readout readability to direct latent *addressability* allowing downstream control via objectives written directly over explicit spatial (z_p) and appearance (z_f) coordinates without training task heads, reward models, or goal images. Driven zero-shot by model-predictive path integral planning (COMB-MPPI; Appendix F), a frozen COMB world model recovers most of the fully trained policy's return on OCRL tasks. Unlike goal-conditioned planners that require visual target images and struggle with episode-dependent relational tasks (e.g., "reach the odd-one-out" or "reach A while avoiding B"), and unlike unstructured slot models (e.g., SOLD) whose latent dimensions lack explicit alignment, named coordinates allow rich, programmatic constraints to be specified declaratively. This opens a promising path toward zero-shot, reward-free control directly from mathematical or symbolic task descriptions.

Limitations. COMB's particles are two-dimensional: image-plane position, scale, and a depth that orders layers rather than measuring distance. This suffices for our fixed-camera 3D suites, but it factorizes the image rather than the scene. Distance to the camera has no independent coordinate and moves image position and scale together, meaning that objects distinct in the physical world can share a particle coordinate. Crucially, this 2D projection suffers from occlusion problems, where overlapping objects become fundamentally entangled or lost entirely from the state representation despite remaining physically distinct. Tasks that depend on full 3D spatial reasoning, such as complex insertion or stacking under heavy occlusion, are therefore outside our current scope. Recent self-supervised 3D object-centric frameworks, such as 3D-DLP (Zhang et al., 2026) and Particle-Splat (He et al., 2026), offer robust solutions for extracting explicit 3D volumetric and geometric representations. Integrating these three-dimensional particle formulations into the COMB model family represents a highly promising direction for extending object-centric policy learning to the full complexity of the physical world.

AI Disclosure and Code Availability

In this work, we used generative AI tools for implementing methods. We have not used generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating or proving mathematical claims, proposing hypotheses, designing experiments. Additionally, we used generative AI tools for editing and revising the manuscript to improve readability, refactoring software code, and searching for or retrieving relevant literature. We have reviewed all AI-assisted work: LLM-generated code implementation and refactoring were thoroughly tested, executed, and verified for correctness by the authors, and all text revisions and literature references were manually inspected for technical accuracy and proper attribution. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI. The codebase of COMB and dosing mechanics is provided as supplementary material with this submission.

References

Jack Brady, Roland S. Zimmermann, Yash Sharma, Bernhard Schölkopf, Julius von Kügelgen, and Wieland Brendel. Provably learning object-centric representations. In *International Conference on Machine Learning*, 2023.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 4603–4623. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arxiv.1901.11390*, 2019.

Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

Tal Daniel and Aviv Tamar. Unsupervised image representation learning with deep latent particles. In *International Conference on Machine Learning*, pp. 4644–4665. PMLR, 2022.

Tal Daniel and Aviv Tamar. Ddlp: Unsupervised object-centric video prediction with deep dynamic latent particles. *Transactions on Machine Learning Research*, 2024.

Tal Daniel, Carl Qi, Dan Haramati, Amir Zadeh, Chuan Li, Aviv Tamar, Deepak Pathak, and David Held. Latent particle world models: Self-supervised object-centric stochastic dynamics modeling. In *International Conference on Learning Representations*, 2026.

Andrea Dittadi, Thomas Kipf, Sjoerd van Steenkiste, Francesco Locatello, Olivier Bachem, and Ole Winther. Generalization and robustness implications in object-centric learning. *arXiv preprint arXiv:2202.00650*, 2022.

Fan Feng, Phillip Lippe, and Sara Magliacane. Learning interactive world model for object-centric reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, and Sai Rajeswar. Representing positional information in generative world models for object manipulation. *arXiv preprint arXiv:2409.12005*, 2024.

Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Focus: object-centric world models for robotic manipulation. *Frontiers in Neurorobotics*, 19:1585386, 2025a.

Stefano Ferraro, Akihiro Nakano, Masahiro Suzuki, and Yutaka Matsuo. When object-centric world models meet policy learning: From pixels to policies, and where it breaks. *arXiv preprint arXiv:2511.06136*, 2025b.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International conference on machine learning*, pp. 2424–2433. PMLR, 2019.

Klaus Greff, Raphael Kaufman, Rishabh Kabra, Elias Dyer, Yoshua Bengio, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025a. doi: 10.1038/s41586-025-08744-2.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025b.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Dan Haramati, Tal Daniel, and Aviv Tamar. Entity-centric reinforcement learning for object manipulation from pixels. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uDxeSZ1wdI.

Lyuxing He, Daniel Guo, Elizabeth Terveen, Deepak Pathak, David Held, and Tal Daniel. Particle-splat: Self-supervised object-centric latent particle splatting. *arXiv preprint arXiv:2609.19463*, 2026.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. DARLA: Improving zero-shot transfer in reinforcement learning. In *International Conference on Machine Learning*, 2017.

Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn. Scalor: Generative world models with scalable object representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJxrKgStDH.

Nikolaus Kapl, Andrea Dittadi, Francesco Locatello, Lars Buesing, and Thomas Kipf. Object-centric representations generalize better compositionally with less compute. In *International Conference on Learning Representations*, 2025.

Thomas Kipf, Sjoerd van Steenkiste, Matthias Minderer, Lars Buesing, and Francesco Locatello. Conditional object-centric learning from video. In *International Conference on Learning Representations*, 2022.

Jannik Kossen, Karl Stelzner, Marcel Hussing, Claas Voelcker, and Kristian Kersting. Structured object-aware physics prediction for video modeling and planning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1e-kxSKDH.

Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive decoders for latent variables identification and cartesian-product extrapolation. In *Advances in Neural Information Processing Systems*, 2023.

Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 6140–6149. PMLR, 13–18 Jul 2020.

Samuel Lippl and Kim Stachenfeld. When does compositional structure yield compositional generalization? a kernel theory. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, 2019.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, A. Mahmood, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, 2020.

Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Aaron Courville, and Sai Rajeswar. GenRL: Multimodal-foundation world models for generalization in embodied agents. In *Advances in Neural Information Processing Systems*, 2024.

Russell Mendonca, Oleh Rybkin, Kostas Daniilidis, Danijar Hafner, and Deepak Pathak. Discovering and achieving goals via world models, 2021.

Cristian Meo, Mircea Tudor Lică, Zarif Ikram, Akihiro Nakano, Vedant Shah, Aniket Rajiv Didolkar, Dianbo Liu, Anirudh Goyal, and Justin Dauwels. Masked generative priors improve world models sequence modelling capabilities. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025. URL https://openreview.net/forum?id=vRdn74Jkqp.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

Milton Llera Montero, Casimir JH Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey Bowers. The role of disentanglement in generalisation. In *International Conference on Learning Representations*, 2021.

Malte Mosbach, Jan Niklas Ewertz, Angel Villar-Corrales, and Sven Behnke. SOLD: Slot object-centric latent dynamics models for relational manipulation learning from pixels. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=XOUpHJPYRX.

Akihiro Nakano, Masahiro Suzuki, and Yutaka Matsuo. Interaction-based disentanglement of entities for object-centric world models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JQc2VowqCzz.

Shukrullo Nazirjonov, Sai Prasanna, Anna Manasyan, and Georg Martius. Better slots, better worlds: Representation quality & robustness in object-centric world models. *arXiv preprint arXiv:2608.12078*, 2026.

Yosuke Nishimoto and Takashi Matsubara. Object-centric world models for causality-aware reinforcement learning. *arXiv preprint arXiv:2511.14262*, 2025.

Carl Qi, Dan Haramati, Tal Daniel, Aviv Tamar, and Amy Zhang. EC-Diffuser: Multi-object manipulation via entity-centric behavior generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=o3pJU5QCtv.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TdBaDGCpjly.

Rubinstein et al. OCAAM: Object-centric attention via applied masking. *arXiv preprint arXiv:2411.00222*, 2025.

Lukas Schott, Julius von Kügelgen, Frederik Träuble, Peter Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. Visual representation learning does not generalize strongly within the same domain. In *International Conference on Learning Representations*, 2022.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning. In *International Conference on Learning Representations*, 2023.

Sjoerd van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are disentangled representations helpful for abstract visual reasoning? In *Advances in Neural Information Processing Systems*, 2019.

Rishi Veerapaneni, John D. Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100, pp. 1439–1456. PMLR, 2020.

Jiayu Wang, Chuxiong Hu, Yunan Wang, and Yu Zhu. Dynamics learning with object-centric interaction networks for robot manipulation. *IEEE Access*, 9:68277–68288, 2021. doi: 10.1109/ACCESS.2021.3077117.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Thaddäus Wiedemer, Jack Brady, Alexander Panfilov, Attila Juhos, Matthias Bethge, and Wieland Brendel. Provable compositional generalization for object-centric learning. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Grady Williams, Andrew Aldrich, and Evangelos A. Theodorou. Model predictive path integral control: From theory to parallel computation. volume 40, pp. 344–357. American Institute of Aeronautics and Astronautics, 2017.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. In *The Eleventh International Conference on Learning Representations*, 2023.

Jaesik Yoon, Yi-Fu Wu, Heechul Bae, and Sungjin Ahn. An investigation into pre-training object-centric representations for reinforcement learning. In *International Conference on Machine Learning*, pp. 40147–40174. PMLR, 2023.

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Self-supervised visual reinforcement learning with object-centric representations. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=xppLmXCbOw1`.

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. *Advances in Neural Information Processing Systems*, 36, 2023.

Ellina Zhang, Madhavan Iyengar, Amir Zadeh, Chuan Li, David Held, Deepak Pathak, and Tal Daniel. 3d-dlp: Self-supervised 3d object-centric scene representation learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. Storm: Efficient stochastic transformer based world models for reinforcement learning. *Advances in Neural Information Processing Systems*, 36:27147–27166, 2023.

Weipu Zhang, Adam Jelley, Trevor McInroe, Amos Storkey, and Gang Wang. Object-centric world models from few-shot annotations for sample-efficient reinforcement learning. *arXiv preprint arXiv:2501.16443*, 2025.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning*, 2025.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

A Extended Related Work

Table 2 situates COMB relative to the models discussed in Section 2. Factorized per-entity codes (SCALOR (Jiang et al., 2020), G-SWM (Lin et al., 2020), Nakano et al. (2023), DLP and DDLP (Daniel & Tamar, 2022; 2024)) have been used with model-free goal-conditioned RL (Zadaianchuk et al., 2021; Haramati et al., 2024), behaviour generation from demonstrations (Qi et al., 2025), and goal-conditioned imitation (Daniel et al., 2026). To our knowledge, COMB is the first to put this type of representation inside an action-conditioned world model trained with online RL. That combination is what lets us study the representation format while the policy is learning. Transformer-based latent world models (Robine et al., 2023; Chen et al., 2022; Meo et al., 2025) instead retain the holistic state used by the methods in Section 2.

Model	**Entities**	**Named attr.**	**Act.-cond.**	**Behaviour**	**Task given as**
DreamerV3 (Hafner et al., 2025a)	–	–	✓	online RL	reward
TD-MPC2 (Hansen et al., 2024)	–	–	✓	online RL	reward
DINO-WM (Zhou et al., 2025)	–	–	✓	planning	goal image
GenRL (Mazzaglia et al., 2024)	–	–	✓	imagination	language, video
SOLD (Mosbach et al., 2025)	slots	–	✓	online RL	reward
SMORL (Zadaianchuk et al., 2021)	objects	✓	–	online RL	goal image
ECRL (Haramati et al., 2024)	particles	✓	–	online RL	goal image
EC-Diffuser (Qi et al., 2025)	particles	✓	–	imitation	goal image
DDLP (Daniel & Tamar, 2024)	particles	✓	–	–	–
LPWM (Daniel et al., 2026)	particles	✓	✓	imitation	image, language
COMB (ours)	particles	✓	✓	online RL	reward, objective

Table 2: Positioning of COMB. *Named attr.*: each entity carries explicit position, scale and appearance coordinates. *Act.-cond.*: a dynamics model conditioned on actions (SMORL and ECRL are model-free). *Behaviour*: how behaviour is learned. DINO-WM and GenRL learn from offline data. COMB is the only entry combining named per-entity attributes with an action-conditioned world model and online RL; its named coordinates also let a task be given as a written objective instead of a reward (Appendix F).

Measuring rather than switching. Related work measures representational structure without retraining, for example with linear probes and disentanglement metrics (Dittadi et al., 2022). Work on compositional generalization instead emphasizes the *coverage* of attribute combinations (Wiedemer et al., 2023; 2024; Lippl & Stachenfeld, 2025). Our setting adds a different kind of control: an invertible family of maps that keeps the information fixed while changing the representation format continuously. We can therefore vary the amount of entanglement rather than switch between two discrete representations.

B Environments

B.1 OCRL

The OCRL suite (Yoon et al., 2023) renders 2D scenes of coloured shapes that a circular agent moves between with discrete actions; reward is sparse and paid on contact with the target. *OCRL-Reach* asks the agent to touch a target square, with the position $\times$ colour split described below. *OCRL-OOO* and *OCRL-OOO-Hard* ask for the object that differs from the others in one attribute, over 2 colours $\times$ 2 shapes and 5 colours $\times$ 5 shapes respectively; OOO-Hard holds out 6 of its 25 (shape, colour) combinations.

Position $\times$ colour split of OCRL-Reach. The target is always a square, coloured blue, green, yellow or pink, and spawned in one of the four arena quadrants. Each episode therefore corresponds to one cell of a 4×4 (colour, quadrant) grid. The three distractors are white or grey triangles and four-pointed stars, so their appearance pool is disjoint from that of the target. The held-out cells form the diagonal, with one held-out cell for each colour and each quadrant. We use two training sets,

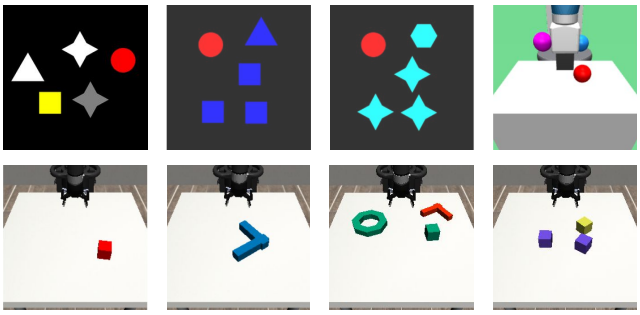


Figure 4: Overview of all environments used in our experiments. Top (left to right): OCRL-Reach, OCRL-OOO, OCRL-OOO-Hard, MOF-Reach. Bottom (left to right): GA-Cube, GA-ShapeColor, GA-Distractor, GA-OOO.

both satisfying the chain condition in Proposition 1: all 12 off-diagonal cells, and a 7-cell subset that forms the minimal chain ($P+F-1$). Table 1 uses the 12-cell split; every model is trained end to end from scratch and never observes a held-out cell. The dose experiments in Section 7 use the 7-cell split and freeze a representation pretrained only on those in-distribution cells. The actor, critic, and reward head of each dose arm are then trained from scratch on the same 7 cells.

B.2 Generalization Arena

We extend the Robosuite Lift environment (Zhu et al., 2020) to support multiple object types. Each object is parameterized by shape, colour, size and *spawn region*: shapes are cube, ring and L-profile, colours come from a palette each variant specifies below, and spawn regions are halves or quadrants of a 0.4×0.4 m area of the table, or all of it. Size is fixed in every task we report. At each episode an object combination is drawn from a predefined set, so held-out combinations can cross appearance with appearance (shape $\times$ colour) or appearance with position (colour $\times$ spawn region). Reward is the dense, staged reward of the Lift task (reaching, grasping, lifting) and is highest when the object is grasped and lifted off the table.

Task variants and splits. *GA-Cube* contains a single red or blue cube spawned in one of four quadrants of the spawn area. The 2×4 (colour, quadrant) grid has two held-out cells, red top-left and blue bottom-right; the remaining six are used for training. Every quadrant appears with at least one colour, and the two colours overlap on seen quadrants, so the training graph satisfies the chain condition of Proposition 1, with one extra cell beyond the minimum $P+F-1=5$. *GA-ShapeColor* draws the single scene object from the 3×3 grid of three shapes and red, blue, and pink. Seven combinations are seen in training and two are held out for evaluation (Figure 5). *GA-Distractor* adds two visually distinct objects to the lift task. The target cube uses four colours (3 ID, 1 OOD); distractors are resampled each episode from a disjoint pool of 4 colours $\times$ 3 non-cube shapes, so their distribution is not itself part of the generalization split. In *GA-OOO*, colour identity is replaced by continuous HSV hue at fixed saturation and value. Three same-size cubes are recoloured each episode, with one differing from the other two. A contiguous hue band is excluded from training and appears in every OOD episode, either as the odd colour or as the shared colour. Memorizing a target colour is therefore insufficient; the policy must use the relation between the objects.



Figure 5: The shape–colour combinations of GA-ShapeColor: the 7 seen in training (left) and the 2 held out for evaluation (right).

C Implementation Details

C.1 DDLP Modules and Inductive Biases

For a given observation $\boldsymbol{x}_t$, DDLP (Daniel & Tamar, 2024) is composed of the following modules:

$$\begin{aligned} \text{Image encoder:} \quad & \boldsymbol{z}_t^{1:K} \sim e_\phi\big(\boldsymbol{z}_t^{1:K} \mid \boldsymbol{x}_t\big)\,, \\ \text{Image decoder:} \quad & \hat{\boldsymbol{x}}_t \sim d_\phi\big(\hat{\boldsymbol{x}}_t \mid \boldsymbol{z}_t^{1:K}\big)\,, \\ \text{Dynamics predictor:} \quad & \boldsymbol{h}_{t+1}^{0:K} = f_\phi\big(\boldsymbol{z}_{0:t}^{0:K}\big)\,, \\ \text{Particle decoder:} \quad & \hat{\boldsymbol{z}}_{t+1}^{0:K} \sim d_\phi^p\big(\hat{\boldsymbol{z}}_{t+1}^{0:K} \mid \boldsymbol{h}_{t+1}^{0:K}\big)\,, \end{aligned} \tag{7}$$

trained by optimizing a modified evidence lower bound (ELBO) built on the Chamfer distance between predicted and encoded particles. Three architectural components produce the particle structure of Eq. 3 without attribute supervision: a spatial-softmax module over convolutional feature maps localizes each particle ($\boldsymbol{z}_p$); a spatial transformer crops a glimpse around that position at the inferred scale ($\boldsymbol{z}_s$), which a convolutional encoder compresses into the appearance features ($\boldsymbol{z}_f$); and a second spatial transformer stitches the decoded glimpses back onto the canvas, with depth ($\boldsymbol{z}_d$) and visibility ($\boldsymbol{z}_v$) governing layering and alpha-compositing over the background. The dynamics predictor, a Particle Interaction Transformer (PINT), predicts the particles forward, which is what lets the world model imagine the consequences of action sequences in latent space.

C.2 COMB Architecture

Perceptual backbone. The DLP encoder processes 64×64 RGB observations using a convolutional backbone with channel widths $[32, 64, 128]$ for foreground glimpses and $[16, 32, 64]$ for the background. We use $K{=}4$ foreground particle slots plus one dedicated background slot, for a total of $N{=}5$ particles per timestep. Each particle carries an appearance feature of dimension $F{=}12$, a 2D position $z_p \in \mathbb{R}^2$, a 2D scale logit $z_s \in \mathbb{R}^2$, a scalar depth logit $z_d \in \mathbb{R}$, and a scalar visibility score $z_v \in [0, 1]$. Glimpses are extracted at patch resolution 8×8 pixels with anchor scale $a_s{=}0.25$. The top-$k{=}3$ most activated encoder positions initialize the prior particle proposals. A variance-based slot ranking heuristic (Daniel & Tamar, 2024) is used to recycle inactive slots, with non-maximum suppression of radius $\delta{=}0.125$ to prevent duplicate detections.

Input token construction. Before entering the policy heads, each particle's raw attributes are assembled into a unified token following Eq. 8:

$$p_t^k = \big[z_{p,t}^k,\ \sigma(z_{s,t}^k),\ z_{v,t}^k,\ \sigma(z_{d,t}^k),\ z_{f,t}^k\big] \in \mathbb{R}^{d_p}, \tag{8}$$

where $\sigma(\cdot)$ is the element-wise sigmoid, so the scale and depth logits are bounded to $(0, 1)$. With $F{=}12$, each token has dimension $d_p = 2{+}2{+}1{+}1{+}12 = 18$. For the actor and critic aggregators, the visibility score $z_{v,t}^k$ is additionally extracted from Eq. 8 before projection and used to build a key-padding mask: particles with $z_{v,t}^k < \epsilon{=}0.1$ are masked from all attention operations, so the effective per-particle projection dimension is $d_p^{\text{eff}} = 17$. The reward head receives the full 18-dimensional token without masking.

Per-suite token geometry. The figures above describe the GA configuration, used on GA-Cube. Token shape is a per-suite hyperparameter, and because the coupling flow splits each token in half (Section 6) it sets the width d the mixer acts on, so we state the geometry of every suite the dial is run on: **GA-Cube** has $d_p{=}18$ (split $d{=}9$), **OCRL-Reach** and **OCRL-OOO-Hard** use $K{=}12$, $d_p{=}10$ (split $d{=}5$, with $F{=}4$), and **MOF-Reach** $K{=}10$, $d_p{=}14$ (split $d{=}7$, with $F{=}8$). *The same λ therefore realizes a different η on each suite*, which is why every dose reported for a suite is measured on that suite's own token cache rather than transferred.

Action-conditioned dynamics predictor. PINT has 6 causal self-attention layers with 8 heads and model dimension 256. Relative Positional Biases (RPB) encode temporal ordering along the context window. Actions are incorporated by concatenating the action vector to each particle token before the first PINT layer.

Particle aggregator. The EIT aggregator architecture is described in detail in Appendix C.3. The actor and critic each instantiate an *independent* aggregator with n_{SA}=1 self-attention layer, 8 heads, embedding dimension d_m=64, FFN width d_h=256, output dimension d_{out}=256, and n_q=2 learned readout queries. The reward head uses the same hidden dimensions but n_{SA}=2 and n_q=1. The three aggregators share no parameters; the representation they read is protected from reward learning by the stop-gradient of Eq. 4 (Appendix C.4).

A summary of all architectural hyperparameters is given in Table 3.

Table 3: Architectural hyperparameters of COMB (GA configuration; per-suite token geometry in the text).

Hyperparameter	**Value**
Perceptual backbone (DLP)	
Image resolution	64×64 RGB
Foreground particle slots K	4
Background slot	1
Appearance feature dim F	12
Particle token dim d_p	18
Foreground encoder channels	[32, 64, 128]
Background encoder channels	[16, 32, 64]
Patch size	8×8
Anchor scale a_s	0.25
Top-k prior proposals	3
NMS radius δ	0.125
Dynamics predictor (PINT)	
Layers	6
Attention heads	8
Model dimension	256
Positional bias	RPB
Action conditioning	token concatenation
EIT aggregator: actor & critic	
Self-attn layers n_{SA}	1
Heads	8
Embedding dim d_m	64
FFN width d_h	256
Output dim d_{out}	256
Readout queries n_q	2
Skip connection	No
Slot masking threshold ϵ	0.1
EIT aggregator: reward head	
Self-attn layers n_{SA}	2
Heads	8
Embedding dim d_m	64
FFN width d_h	256
Output dim d_{out}	256
Readout queries n_q	1
Skip connection	Yes (gated)

C.3 EIT AGGREGATOR

The Entity Interaction Transformer (EIT) aggregator (Haramati et al., 2024) maps an unordered set of particle tokens to a fixed-size summary $\mathbf{g}_t \in \mathbb{R}^{d_{\text{out}}}$ consumed by a downstream head (Figure 6). Its first operation is a shared linear projection of each token of Eq. 8,

$$h_t^k = W_{\text{proj}}\, p_t^k + b_{\text{proj}} \in \mathbb{R}^{d_m}, \tag{9}$$

the layer that absorbs any linear scramble of the tokens (Section 6). For the actor and critic the visibility score is removed first, so $W_{\text{proj}} \in \mathbb{R}^{d_m \times 17}$; for the reward head $W_{\text{proj}} \in \mathbb{R}^{d_m \times 18}$. *Stage 1* applies n_{SA} pre-norm transformer blocks (multi-head self-attention followed by a three-layer ReLU feed-forward network of width d_h) with no positional encoding, so it is permutation-equivariant and each output encodes a particle's relations to the other visible particles; particles with $z_v^k < \epsilon$ are excluded by a key-padding mask. *Stage 2* pools the set with n_q learned readout queries (initialized with Gaussian noise, σ=0.02) through one cross-attention block; the queries take no part in Stage 1. The n_q readouts are flattened, layer-normalized and passed through a two-layer ReLU MLP to give $\mathbf{g}_t$. For the reward head, a gated skip connection adds a mean-pooled projection of the raw tokens, which gives a stable gradient path early in training.

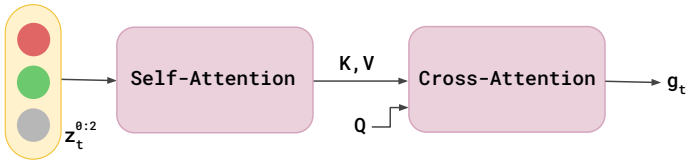


Figure 6: The EIT aggregator: self-attention over the particle tokens, then cross-attention from learned readout queries.

C.4 WHY THE REWARD GRADIENT IS STOPPED

Section 4 stops the reward gradient at the aggregator input (Eq. 4). We explain that choice here; we do not ablate it.

The rewards of our tasks are functions of relations between entities: the distance from the agent or gripper to its target, and on GA whether the gripper holds the cube and how high it lifts it. A reward head that reads a set of particles must combine at least two tokens to predict such a quantity, which in COMB is the job of the aggregator's attention. If the reward loss also trained the encoder, it would open an easier route: writing the relation into a single particle, for instance letting the gripper's particle carry the cube's offset from it, so that one token predicts the reward on its own. That route runs against the property this paper studies. A particle that carries information about two objects is no longer a sum of one entity's attributes: a change to one object then moves the token of another, and the localization of Section 3, by which an unseen attribute value perturbs only its own coordinates, no longer holds across entities.

With the stop-gradient, the encoder and the dynamics predictor are trained only by reconstruction and prediction (Appendix C.5), nothing rewards a particle for describing more than its own entity, and every computation across entities is confined to the aggregator, at whose input the dial of Section 6 intervenes. The architecture names the coordinates, but the binding of each particle to one entity is something training has to protect. We did not train a variant in which the reward gradient reaches the encoder, so how much erosion the stop-gradient prevents is untested; such a variant would manipulate format from inside training, complementary to the dial, which manipulates it after the encoder is frozen.

C.5 TRAINING DETAILS

Table 4 summarizes training hyperparameters for all methods. All models are trained on 64×64 RGB observations with action repeat 2 and a replay buffer of 500 k transitions. World model and policy updates are performed every 5 environment steps.

Imagination-based policy training. Policy optimization uses $S{=}8$ stride starting positions per sequence, yielding $B{\times}S = 64{\times}8 = 512$ parallel imagined rollouts per imagination call over a horizon of $H{=}10$ steps. Starting offsets are sampled independently at random per batch element and stride, rather than a fixed offset shared across the batch. The actor and critic are updated for $E{=}1$ epoch per imagination call.

Loss coefficients. The DLP ELBO is trained with reconstruction weight $\beta_{\text{rec}}{=}1.0$, KL weight $\beta_{\text{KL}}{=}0.1$ (encoder–prior KL balance $\kappa{=}0.001$), dynamics prediction weight $\beta_{\text{dyn}}{=}1.0$ (dynamics KL balance $\kappa_{\text{dyn}}{=}0.5$), and reward weight $\beta_{\text{rw}}{=}1.0$. Because of the stop-gradient of Eq. 4, no reward gradient reaches the encoder or the dynamics predictor (Appendix C.4).

Table 4: Training hyperparameters.

Hyperparameter	**COMB**	**DreamerV3**	**TD-MPC2**	**SOLD**
Optimization				
Batch size	64	64	256†	64
Sequence length	15	15	–	15
Imagination horizon H	10	10	5†	10
Updates per 5 env. steps	1	1	1	1
Imagined strides S	8	–	–	–
Policy update epochs E	1	1	–	1
Optimizers & learning rates				
Actor LR	10^{-4}	$3{\times}10^{-5}$	10^{-4}†	$3{\times}10^{-5}$
Critic LR	10^{-4}	$3{\times}10^{-5}$	10^{-4}†	10^{-4}
World model LR	$2{\times}10^{-4}$	10^{-4}	10^{-4}†	10^{-4}
All optimizers	Adam	Adam	Adam	Adam
Reward and returns				
Discount γ	0.99	0.99	0.99†	0.99
λ-return	0.95	0.95	–	0.95
Slow-target update	100 steps	100 steps	–	100 steps
Entropy bonus α	0.005	–	–	–
Actor gradient estimator	dyn. backprop.	dyn. backprop.	–	dyn. backprop.
COMB-specific loss weights				
β_{rec}	1.0	–	–	–
β_{KL}	0.1	–	–	–
β_{dyn}	1.0	–	–	–
β_{rw}	1.0	–	–	–
KL balance κ	0.001	–	–	–
Dyn. KL balance κ_{dyn}	0.5	–	–	–

† Default TD-MPC2 value for `model_size=12`.

Online MPPI planner. Action selection, both during data collection and at evaluation, is performed by the TD-MPC2-style planner of Section 4, which scores rollouts with the learned reward head and a bootstrapped slow-target critic value. This *differs* from the zero-shot latent-MPPI planner evaluated in Appendix F, which is reward-free and purely analytical. The parameters in Table 5 are shared by every dosing arm and every suite. On the discrete actions of OCRL the proposal is a categorical distribution sampled with the Gumbel-max trick, so the action standard deviation and bounds apply to the continuous suites only.

Table 5: Online MPPI planner hyperparameters (training & evaluation action selection).

Hyperparameter	Value
Planning horizon	3
Candidate sequences	64
Policy-seeded trajectories	8
Refinement iterations	3
Elites	16
Temperature τ	0.5
Action std range	$[0.05,\ 1.0]$
Action bounds	$[-1.0,\ 1.0]$

C.6 Baseline Configurations

DreamerV3. We use the DreamerV3 implementation of Hafner et al. (2025a) with the reduced-capacity `small` configuration, sized to a parameter count comparable to COMB's. The RSSM uses a deterministic recurrent state of dimension 1184, stochastic discrete latents with 24×24 categories, and a GRU hidden size of 256. The CNN image encoder has depth 24 with kernel sequence $[4, 4, 4, 4]$ and the CNN decoder uses $[5, 5, 6, 6]$; both use layer normalization. The actor and critic are three-layer MLPs with 448 units per layer. The world model is trained end to end with an image decoder and a reward head. The actor gradient estimator is the straight-through dynamics gradient. Optimizers are Adam with learning rate 3×10^{-5} for actor and critic and 10^{-4} for the world model.

TD-MPC2. We use the official TD-MPC2 implementation of Hansen et al. (2024) with `model_size=12` (approximately 12 M parameters), operating on 64×64 RGB observations. All other hyperparameters follow the default configuration of the public repository.

SOLD. We use the authors' official implementation of SOLD (Mosbach et al., 2025), an object-centric world model designed independently of this work: Slot-Attention perception over image tokens, action-conditioned latent dynamics in slot space, and reward, actor and critic heads that aggregate the slots and are trained in imagination. It separates the scene into entities, but each slot is an unstructured embedding rather than a set of named per-attribute coordinates, so it is object-centric but not factorized. We match its parameter budget to COMB and train it on the same 64×64 RGB observations, update cadence, and replay budget as all other methods.

D Theory: Proofs, a Graded Bound, and the Offline Test

We prove Proposition 1 here, then examine how its guarantee changes between the linear and fully nonlinear cases. We also give the complete protocol for the offline test in Section 7.1, including the within-attribute control.

Setup. Position takes values $p \in \{1, \dots, P\}$ and appearance $f \in \{1, \dots, F\}$. The factorized code is $z(p, f) = e_p + c_f$ (Eq. 1) and a readout is $h(z) = w^\top z + b$. To state the chain condition, draw a graph with one node for each position value, one node for each appearance value, and an edge for each seen combination $(p, f) \in \mathcal{S}$. The condition is that this graph is connected. A connected graph on $P{+}F$ nodes has at least $P{+}F{-}1$ edges, and any spanning tree has exactly that many.

Proof of (a). Take two readouts that agree on $\mathcal{S}$ and call their difference δ. On the factorized code it splits as $\delta(p, f) = u_p + v_f$, with $u_p = \Delta w^\top e_p + \Delta b$ and $v_f = \Delta w^\top c_f$. Agreement on a seen combination means $u_p = -v_f$ for that pair, so each edge of the graph fixes one endpoint's value from the other. On a connected graph this forces $u_p = t$ for every p and $v_f = -t$ for every f, for a single constant t. Hence $\delta(p, f) = 0$ for all $P\times F$ combinations. Equivalently, every unseen combination is reached by applying Eq. 2 repeatedly along the chain. No condition on the token dimension is needed.

Proof of (b). Let $\psi(z) = Mz + m$ with M invertible. Then $w^\top \psi(z) + b = (M^\top w)^\top z + (w^\top m + b)$, and as (w, b) ranges over all readouts, so does $(M^\top w,\ w^\top m + b)$. Readouts of $\psi(z)$ are exactly the readouts of z, so part (a) applies unchanged, whatever the conditioning of M. The same substitution works for any head whose first layer is a trainable linear map, which is why part (b) covers our transformer heads: the set of functions they can compute does not change, although a badly conditioned M can still slow optimization.

Proof of (c). Suppose the $P \times F$ scrambled codes, each extended with a constant entry 1, are linearly independent, which is possible once the token dimension is at least PF. Then for any target values on the grid some readout hits them exactly. Choose two sets of values that coincide on $\mathcal{S}$ and differ arbitrarily everywhere else: both are realized, so agreement on $\mathcal{S}$ says nothing about the rest, and only seeing every combination does. This is the extreme case. Our tokens have fewer dimensions than grid cells and our mixers are far from generic, which is why we measure η rather than assume either end.

Between the two ends: a graded bound. Write any mixer as its best affine fit plus a residual, $\psi(z) = Mz + m + r(z)$, and let $\varepsilon = \max_{(p,f)} \|r(z(p, f))\|$ be the largest residual on the grid. Bound the readout weights, $\|w\| \le \rho$, as weight decay and early stopping do in practice. For two readouts that agree on $\mathcal{S}$, split their difference as $\delta = \delta_{\text{lin}} + \delta_{\text{res}}$. The affine part gives δ_{lin}, which is additive in (p, f) exactly as in part (a), and the residual part satisfies $|\delta_{\text{res}}| \le 2\rho\varepsilon$ everywhere. On seen combinations $\delta = 0$, so $|\delta_{\text{lin}}| \le 2\rho\varepsilon$ there. Any unseen combination is joined to the seen ones by a path of at most $P+F-1$ edges, and adding and subtracting the additive values along that path bounds $|\delta_{\text{lin}}|$ by $(P+F-1)\, 2\rho\varepsilon$. Together,

$$\left|\delta(p^\star, f^\star)\right| \ \le\ 2\,(P+F)\,\rho\,\varepsilon. \tag{10}$$

The bound is zero when the residual is zero, which recovers part (b), and grows only in proportion to the residual, so small departures from linearity allow only small disagreement on unseen combinations. The dose of Eq. 5 summarizes the same residual as a share of variance rather than as its maximum, so we do not claim an exact relation between η and ε; the claim we test is that cost rises with η.

The offline test: protocol. We use the active particle tokens (visibility above 0.1) of 12 OCRL-OOO-Hard episodes, encoded by the frozen 100k-step COMB snapshot, and standardize every coordinate. The position block is $\boldsymbol{z}_p$ (2 dimensions) and the appearance block is $\boldsymbol{z}_f$ (4 dimensions on this suite); scale, depth and visibility belong to neither. Each block's leading principal component is cut at its quartiles, giving $P{=}F{=}4$ levels and 16 combinations. For each seed we hold out 4 combinations at random, redrawing until the remaining 12 satisfy the chain condition, and set the target to $y = w_p^\top \boldsymbol{z}_p + w_f^\top \boldsymbol{z}_f$ with fresh Gaussian weights, standardized. The target is additive by construction, so any failure to recover held-out combinations comes from the representation, not the target. Mixers are the coupling flow of Section 6, calibrated on the same tokens, and the placebo is a random orthogonal map. Each dose is measured with the affine fit of Eq. 5; at the level of single tokens that fit has few regressors against many rows, so it does not overfit. The linear head is ridge regression with penalty $10^{-6}n$; the MLP head has two hidden layers of 256 GELU units, trained with Adam (learning rate $3{\times}10^{-3}$, 400 epochs, cosine schedule). We report excess risk $\mathrm{MSE}/\mathrm{Var}(y)$, averaged over 5 seeds.

What the table supports. Both dose-zero maps recover the held-out combinations exactly on the linear head, which is part (b) observed on real tokens (P1); the placebo rotates every coordinate, so this is not the trivial result of leaving the tokens alone. With the linear head, held-out risk rises at every step in η and exceeds the risk on seen combinations at every nonzero dose, part (c) (P3). The MLP head fits the seen combinations almost perfectly at every dose, and its held-out risk rises overall, from about 0.001 to 0.171, so a more expressive head does not restore transfer (P3). It does not rise at every step: it is lower at $\eta{=}0.122$ than at $\eta{=}0.054$, so for the MLP the claim is a trend rather than strict monotonicity. Two limits apply. The grid is built from principal-component bins of the model's own blocks rather than from ground-truth attributes, so the test checks additivity in the factors the representation actually has; and the target is synthetic.

Mixing within attributes: protocol. To separate nonlinearity from mixing across attributes, we apply the coupling flow of Section 6 ($L{=}6$) to the standardized position block (2 coordinates) and,

		linear head		MLP head	
Mixer	**dose** η	seen	held-out	seen	held-out
identity (COMB)	0.000	0.0000	**0.0000**	0.00000	0.00117
placebo (orthogonal)	0.000	0.0000	**0.0000**	0.00000	0.00095
coupling L=6, λ=0.5	0.054	0.0252	0.0358	0.00002	0.01584
coupling L=2, λ=1.0	0.122	0.0349	0.0601	0.00002	0.00422
coupling L=4, λ=1.0	0.212	0.0444	0.0914	0.00005	0.06084
coupling L=6, λ=1.0	0.267	0.1002	0.1517	0.00010	0.07163
coupling L=6, λ=1.5	0.537	0.2690	0.3788	0.00047	0.17086

Table 6: Excess risk ($\mathrm{MSE}/\mathrm{Var}(y)$, 0 = exact recovery) on seen and held-out attribute combinations, for the offline test of Section 7.1. Mean over 5 seeds, rows sorted by dose. The placebo rotates every coordinate yet has $\eta = 0$, and like the identity it recovers held-out combinations exactly.

as an independent random flow, to the appearance block (4 coordinates), leaving scale, depth and visibility untouched. The code remains a concatenation of per-attribute codes, $(g(\boldsymbol{z}_p), h(\boldsymbol{z}_f))$. Tokens, grid, splits, targets, heads and the dose estimator are those of Table 6. We also report how linearly each attribute can still be read: the held-out R^2 of the best affine map from the mixed token to the original position or appearance block (random half split of the tokens).

		readability R^2		**linear head**		**MLP head**	
Mixer	**dose** η	position	appearance	seen	held-out	seen	held-out
identity (COMB)	0.000	1.00	1.00	0.0000	0.0000	0.0000	0.0012
across attributes (Table 6)							
coupling L=6, λ=0.5	0.054	0.98	0.97	0.0252	0.0358	0.0000	0.0158
coupling L=6, λ=1.0	0.267	0.90	0.89	0.1002	0.1517	0.0001	0.0716
coupling L=6, λ=1.5	0.537	0.83	0.73	0.2690	0.3788	0.0005	0.1709
within attributes							
coupling L=6, λ=0.5	0.010	0.94	0.95	0.0675	0.1290	0.0000	0.0245
coupling L=6, λ=1.0	0.028	0.96	0.92	0.0701	0.1401	0.0001	0.1129
coupling L=6, λ=1.5	0.061	0.54	0.92	0.1160	0.2538	0.0005	0.1813
coupling L=6, λ=2.0	0.046	0.65	0.87	0.1534	0.2617	0.0016	0.6521
coupling L=6, λ=3.0	0.039	0.70	0.85	0.1628	0.2514	0.0045	0.5949
coupling L=6, λ=5.0	0.044	0.54	0.79	0.1952	0.2824	0.0037	0.4333

Table 7: The offline test of Table 6 with the coupling flow applied across attributes (the full token) or within them (position and appearance blocks separately). Readability: held-out affine R^2 from the mixed token to the original block. Excess risk ($\mathrm{MSE}/\mathrm{Var}(y)$) and dose as in Table 6, mean over 5 seeds.

What the within-attribute arms show. The within-attribute perturbations are more costly at comparable doses. At η=0.061 (λ=1.5), the linear head gives a held-out risk of 0.254 ± 0.073, versus 0.036 ± 0.007 for the cross-attribute arm at η=0.054; for the MLP the corresponding values are 0.181 ± 0.087 and 0.016 ± 0.011 (s.e.m. over seeds). The MLP still fits the seen combinations below 10^{-3}. Even the mildest within-attribute arm, at η=0.010, has a linear held-out risk close to the cross-attribute arm at η=0.267. In both families, the common change is a loss of linear readability: position R^2 reaches 0.54 at dose 0.06 for the within-attribute arm, whereas the cross-attribute arm retains 0.98.

This does not conflict with Proposition 1. The proposition treats each attribute value as a single point; on that discrete grid, $(g(\boldsymbol{z}_p), h(\boldsymbol{z}_f))$ remains additive and part (a) applies. Here, however, the attributes vary continuously within each combination and the target is linear in those continuous variables. A linear head can only approximate a bent attribute, and the least-squares fit can trade errors between the two blocks. Those errors may cancel on seen combinations without cancelling on

unseen ones. An MLP can fit the seen combinations much more closely, but it is not constrained to combine the attributes additively. Separate coordinates are therefore necessary for the transfer result in part (a), but they are not sufficient: the relevant coordinates must also remain linearly readable from their own blocks.

There are two further caveats. The scalar dose understates these within-attribute perturbations because only 6 of the 10 token coordinates are modified, with the strongest effect in the two-dimensional position block; in this case the dose is not even monotone in λ. Finally, as in Table 6, this is an offline experiment with a synthetic target. We did not test the within-attribute perturbation online.

What the simple setting leaves out. *Nonlinear heads.* Part (b) remains exact for any head whose first layer is trainable and linear. Parts (a) and (c) give the two limiting cases; the MLP results above and the online experiments show how the intermediate case behaves in practice. *Which attributes.* The DLP appearance vector $\boldsymbol{z}_f$ is a single unstructured bag holding colour, shape, rotation and lighting together, and nothing in a reconstruction objective asks it to separate them. The split the architecture actually provides is position versus appearance, which is the one the theory uses. *Learned, not given.* Equation 1 is assumed, whereas COMB has to learn its factorization and can partly fail to. The reward stop-gradient is designed to protect it during online training (Appendix C.4).

Why a linear scramble is not an ablation. Part (b) rules out a common control. A rotation, random linear map, or piecewise-linear map that is mostly absorbed by the first linear layer does not change the relevant function class and therefore does not test format. Our first control, a LeakyReLU between two random orthogonal matrices, gives $\eta = 0.0068$ on real tokens and a linear reconstruction R^2 of 0.93. We treat it as a placebo, not as a format ablation. Any such ablation should be reported together with its measured dose.

E Entanglement Dosing: Construction and Extended Results

E.1 The Dosing Mixer and Its Controls

Dosing mixer. Each particle token $\boldsymbol{p}_t^k \in \mathbb{R}^{d_p}$ is passed, before the aggregator, through the calibrated coupling flow of Section 6. Concretely: (1) whiten with per-coordinate moments frozen at calibration; (2) clamp to ± 8 standard deviations; (3) apply L coupling layers, each permuting coordinates and updating one half by $\boldsymbol{v} \leftarrow \boldsymbol{v} + s_i t_i(\boldsymbol{u})$ with t_i a frozen two-hidden-layer MLP of width 64; (4) un-whiten and re-match the unmixed token's per-coordinate moments with a fixed diagonal affine, so that no arm differs from another by a pure scale change. Steps (1), (2) and (4) are shared by *every* arm including the placebo and the identity, so the pipeline itself is never a confound. Because mixing spreads the visibility score over other coordinates, the key-padding mask of Appendix C.2 is disabled in every dose arm, the identity included. Gains s_i are set once, on a held-out calibration batch, so that $\mathrm{std}(s_i t_i(\boldsymbol{u})) = \lambda\, \mathrm{std}(\boldsymbol{v})$.

Two knobs define an arm. **Strength** λ and **depth** L both move the realized dose η; that they move it along a *single* axis is what the $L{=}4$ arms of Table 9 verify (Appendix E.2).

Controls. The **placebo** replaces the coupling core with a frozen random orthogonal matrix, while keeping whitening, clamping, and moment matching unchanged. The **linear twins** use linear maps whose singular-value spectra are fitted to match either $\mathrm{cond}(J)$ or $\mathrm{cond}(\mathrm{cov})$ of the coupling arm. For a linear map on whitened input, $\mathrm{cov}_{\mathrm{out}} = W^\top W$, so $\mathrm{cond}(\mathrm{cov}) = \mathrm{cond}(J)^2$ exactly; one linear map therefore cannot match both quantities at once. The coupling flow avoids this constraint because its shear leaves the preserved half's variance unchanged. At $\lambda{=}1$, for example, $\mathrm{cond}(J) = 34.6$ and $\mathrm{cond}(\mathrm{cov}) = 119.6$, whereas a linear map with the same Jacobian conditioning has $\mathrm{cond}(\mathrm{cov}) = 1197$. We therefore include both twins. The pre-specified falsification criterion is simple: if the coupling arm underperforms both twins, the effect cannot be attributed to conditioning on either axis.

Verification. Over the full $N{\cdot}d_p$ scene we measure $|\log\det J| \le 10^{-6}$, a closed-form inverse round-trip error below $4{\times}10^{-6}$, and fewer than 10^{-3} of coordinates clamped, so no information is lost to saturation.

E.2 Extended Dose–Response Analysis

Mixer	dose η	R^2 pos. / col.	ID return	vs. COMB	transfer gap
OCRL-Reach (2D, sparse reward, 100k steps, 7 of 16 cells seen)					
COMB (identity)	0.000	1.00 / 0.76	$\mathbf{70.2} \pm 2.0$	–	-1.5 ± 5.2
placebo (orthogonal)	0.000	1.00 / 0.76	70.2 ± 2.9	0%	-2.1 ± 4.7
linear twin, cov-matched	0.000	1.00 / 0.76	67.9 ± 2.2	−3%	$+4.2 \pm 1.9$
coupling L=6, λ=0.75	0.142	0.91 / 0.68	62.2 ± 2.0	−11%	-16.0 ± 3.3
coupling L=6, λ=1	0.228	0.87 / 0.61	51.5 ± 1.9	−27%	-22.0 ± 3.1
coupling L=6, λ=1.2	0.356	0.83 / 0.53	48.7 ± 1.9	−31%	-26.2 ± 1.9
coupling L=6, λ=1.5	0.629	0.78 / 0.45	44.4 ± 1.8	−37%	-30.2 ± 3.5
OCRL-OOO-Hard (2D, sparse reward, 250k steps, 19 of 25 cells seen)					
COMB (identity)	0.000	1.00 / 0.66	$\mathbf{49.7} \pm 3.7$	–	-2.3 ± 2.7
placebo (orthogonal)	0.000	1.00 / 0.66	52.6 ± 3.7	+6%	-4.0 ± 5.4
linear twin, cov-matched	0.000	1.00 / 0.66	49.5 ± 1.9	0%	-5.1 ± 3.0
linear twin, jac-matched	0.000	1.00 / 0.66	47.3 ± 4.3	−5%	-11.6 ± 4.3
coupling L=6, λ=0.75	0.132	0.94 / 0.66	50.4 ± 1.7	+2%	-24.2 ± 1.8
coupling L=6, λ=1	0.232	0.88 / 0.58	35.9 ± 2.6	−28%	-32.0 ± 3.4
coupling L=6, λ=1.2	0.374	0.83 / 0.51	25.9 ± 3.6	−48%	-42.1 ± 2.7
coupling L=6, λ=1.5	0.656	0.77 / 0.45	19.8 ± 2.2	−60%	-46.2 ± 2.6
GA-Cube (3D, dense reward, 300k steps, 6 of 8 cells seen)					
COMB (identity)	0.000	0.97 / 0.55	$\mathbf{175.0} \pm 3.4$	–	-0.2 ± 1.4
placebo (orthogonal)	0.000	0.97 / 0.55	165.0 ± 3.7	−6%	-2.7 ± 3.1
linear twin, jac-matched	0.000	0.97 / 0.55	167.0 ± 3.7	−5%	-1.4 ± 1.1
coupling L=6, λ=0.75	0.159	0.91 / 0.50	158.0 ± 3.4	−10%	-9.0 ± 2.1
coupling L=6, λ=1	0.203	0.82 / 0.49	144.3 ± 3.7	−18%	-11.4 ± 2.1
coupling L=6, λ=2	0.222	0.63 / 0.42	131.2 ± 4.0	−25%	-20.7 ± 1.4
coupling L=6, λ=3	0.273	0.42 / 0.39	120.2 ± 4.0	−31%	-27.5 ± 3.2

Table 8: Per-arm numbers behind Figure 3. R^2: affine probe from the mixed representation to true position and colour on the suite's frozen snapshot, held out by episode. ID return: mean over each seed's last 25k steps, $\pm$ s.e.m. over 3 seeds (5 on OCRL-OOO-Hard, 4 for its L=6, λ=1.2 arm). Transfer gap: held-out minus in-distribution return at matched in-distribution skill (60–90% of COMB's final), as a percentage of COMB's final return, $\pm$ s.e.m. over seeds. Every arm is an exact bijection of every other.

Protocol and statistics. Figure 3 and Table 8 summarize the three suites with held-out cells. For each seed, in-distribution return is averaged over the final 25k steps of a fixed budget: 100k for OCRL-Reach, 250k for OCRL-OOO-Hard, and 300k for GA-Cube. We use three seeds, except for OCRL-OOO-Hard, which uses five. Held-out return is measured from the same trained agent on the held-out cells at each evaluation point. The training sets contain 7 of 16 (colour, quadrant) cells on OCRL-Reach, 6 of 8 on GA-Cube, and 19 of 25 (shape, colour) combinations on OCRL-OOO-Hard. Dose is measured separately on each suite's frozen tokens, since a fixed λ need not give the same η across token geometries. On the position $\times$ colour suites, the weakest dosed arm falls below the null band at p=0.05 on OCRL-Reach (-11%) and p=0.02 on GA-Cube (-10%), using Welch tests over seeds. The dose-zero arms are unresolved from COMB on all three suites ($p \geq 0.12$). The transfer gap separates dosed from dose-zero arms with p=5×10^{-7}, 3×10^{-5}, and 2×10^{-11} on OCRL-Reach, GA-Cube, and OCRL-OOO-Hard, respectively; correlations with dose are -0.84, -0.89, and -0.86. The full OCRL-OOO-Hard ladder and MOF-Reach in Table 9 provide more low-dose points, both linear twins on one suite, and matched arms of different depth.

The two-axis bracket. On OCRL-OOO-Hard, the two linear twins bracket the coupling arm on the two conditioning axes. The cov-matched twin has the same covariance conditioning but about $3\times$ easier Jacobian conditioning; the jac-matched twin has the same Jacobian conditioning but is $10\times$ harder on covariance conditioning, which is the quantity that affects optimization here. Both twins remain within 5% of COMB, whereas the coupling arm at matched nominal strength loses

Mixer	dose η	Return	vs. COMB
OCRL-OOO-Hard (2D, sparse reward)			
Dose-zero controls			
COMB (identity)	0.000	$\mathbf{49.7} \pm 3.7$	–
placebo (orthogonal)	0.000	52.6 ± 3.7	$+6\%$
linear twin, cov-matched	0.000	49.5 ± 1.9	0%
linear twin, jac-matched	0.000	47.3 ± 4.3	-5%
Dose ladder			
coupling L=4, λ=1.0	0.131	45.1 ± 1.8	-9%
coupling L=6, λ=0.75	0.132	50.4 ± 1.7	$+2\%$
coupling L=6, λ=0.85	0.173	41.2 ± 4.7	-17%
coupling L=6, λ=0.925	0.203	38.0 ± 2.1	-24%
coupling L=4, λ=1.38	0.222	31.6 ± 1.8	-36%
coupling L=6, λ=1.0	0.232	35.9 ± 2.6	-28%
coupling L=6, λ=1.2	0.374	25.9 ± 3.6	-48%
coupling L=6, λ=1.5	0.656	19.8 ± 2.2	-60%
MOF-Reach (3D, dense reward)			
Dose-zero controls			
COMB (identity)	0.000	$\mathbf{66.1} \pm 4.9$	–
placebo (orthogonal)	0.000	64.6 ± 2.9	-2%
linear twin, jac-matched	0.000	64.7 ± 1.9	-2%
Dose ladder			
coupling L=6, λ=0.75	0.096	52.4 ± 5.9	-21%
coupling L=6, λ=1.0	0.174	48.9 ± 5.4	-26%
coupling L=6, λ=1.5	0.416	31.9 ± 5.3	-52%

Table 9: In-distribution return at a fixed 250k-step budget on the full OCRL-OOO-Hard ladder, with the L=4 arms and the intermediate rungs, and on MOF-Reach (3D), a fourth suite without held-out cells. Return is the mean over each seed's last 25k steps, $\pm$ s.e.m. over 5 seeds (4 for OCRL-OOO-Hard L=6, λ=1.2). Every arm is an exact bijection of every other: only the coordinate system differs. On OCRL-OOO-Hard the four dose-zero arms span 47.3–52.6 (-5% to $+6\%$), an empirical null band within which nothing is resolved. Doses are measured on each run's own checkpoint. MOF-Reach's cov-matched twin is omitted because it is not a valid control on this suite (see the MOF-Reach paragraph below); the twins' construction and conditioning are in Appendix E.1.

28%. Neither conditioning statistic orders the four dose-zero arms, whose returns range from 47.3 to 52.6 (-5% to $+6\%$). By the criterion in Section 6, the dose effect is therefore not explained by either conditioning measure. OCRL-Reach and GA-Cube each include one twin; in both cases it stays within the null band while the matched coupling arm loses 27% and 18%, respectively.

The transfer gap does not depend on the band. The comparison band was chosen on OCRL-OOO-Hard after those runs were completed. We therefore repeat the analysis over several bands to check how sensitive the result is to that choice. Over bands of width 30 points, the dose-zero arms lead the dosed arms by 8.3, 14.0, 23.8 and 27.0 points on OCRL-Reach and by 8.3, 13.0, 15.7 and 16.6 on GA-Cube, for bands starting at 40, 50, 60 and 70% of COMB's final return, with correlation to dose over seeds between -0.54 and -0.90. The only unresolved case is the lowest band on OCRL-Reach (p=0.07), where both evaluations sit closest to the reward floor; the wide bands 40–100% and 50–100% give 15.9 and 17.9 points on OCRL-Reach and 11.4 and 12.3 on GA-Cube. On OCRL-OOO-Hard every band between 40% and 100% puts the dosed arms of Figure 3 20–30 points below the controls (r between -0.84 and -0.87), and 19–29 points with the full ladder of Table 9, including the L=4 arms and the λ=0.85 and 0.925 rungs.

Readability is comparable within a suite, not across suites. R^2 normalizes probe error by the variance of the target. A suite in which the task objects occupy less of the image will therefore report a lower R^2 for the same absolute error. This is what separates the 3D suites from the 2D ones: a fixed camera that must see the robot and the table confines the objects' workspace to a small

part of the image, whereas the OCRL arena fills it. At matched dose the absolute error is the same across suites: on a probe whose targets are centred on each frame's mean object position, 0.1435 on OCRL-OOO-Hard at η=0.37 against 0.1438 on MOF-Reach at η=0.42 (DLP units), while MOF-Reach's task objects span about $2.5\times$ less of the image (0.16 against 0.39), which is why its R^2 falls to 0.16 where OCRL-OOO-Hard's stays at 0.86. Return, as a share of each suite's COMB, follows the absolute error as one relation across both suites (r=-0.89 over all 18 arms, -0.97 over the 11 dosed arms) where it follows R^2 only within each (r=0.56 pooled). The same holds for GA-Cube against the OCRL suites on the absolute probe of Figure 3a: at matched dose ($\eta\approx0.20$) the mixer displaces position by the same absolute error on both (0.14 in DLP units), while the true positions spread 0.25 on GA-Cube against 0.41 on OCRL. We also tested the natural alternative, that a 2D code writes a 3D position through position and scale jointly and is therefore more fragile under mixing. It is not supported: the probe's target is the object's projected image position, the projection adds only a constant floor (MOF-Reach's COMB reads position at 0.95), depth and scale carry about 0.1% of the token variance on OCRL-OOO-Hard and MOF-Reach and under 1% on GA-Cube, and on GA-Cube an object's inferred scale correlates only weakly with its image position ($|r| \leq 0.21$). GA-Cube also shows that η, a variance share over the whole token, can understate the damage to the attribute a task reads: from λ=2 to 3 its η rises by only 0.05 while position readability falls from 0.63 to 0.42.

The shape of the curve at low dose. On doses measured from each run's own checkpoint, the last OCRL-OOO-Hard arm inside the null band is η=0.13 (50.4, $+2\%$). At η=0.17 return falls below the band without being resolved from COMB (-17%, p=0.20, Welch over seeds), and the first resolved deficit is at η=0.20 (-24%, p=0.03). Past it the L=6 ladder declines steadily: -28% at 0.23, -48% at 0.37 and -60% at 0.66. MOF-Reach, with three dosed arms, shows the same monotone decline (-21%, -26%, -52% at η=0.10, 0.17, 0.42). OCRL-OOO-Hard is the only suite on which a dosed arm stays inside the null band: on MOF-Reach, OCRL-Reach and GA-Cube the weakest dosed arm already falls below it, so we read the in-band arm as a cost too small to resolve at five seeds rather than as a threshold below which the head absorbs the scramble.

Depth is controlled at two points. We pair each L=4 arm with the closest L=6 arm in measured dose. This lets us vary depth while keeping the realized dose nearly fixed: L=$4, \lambda$=1.0 against L=$6, \lambda$=0.75 at η=0.131 and 0.132 (45.1 ± 1.8 against 50.4 ± 1.7, p=0.06), and L=$4, \lambda$=1.38 against L=$6, \lambda$=1.0 at η=0.222 and 0.232 (31.6 ± 1.8 against 35.9 ± 2.6, p=0.22). Neither pair is resolved, so "more layers hurt more" is not supported, and if anything the shallower map costs slightly more. Readability accounts for that residual: at matched dose the L=4 arms read true position slightly worse (0.92 against 0.94, 0.84 against 0.88) and far worse when the probe is scored on (quadrant, colour) cells it was not fitted on (0.76 against 0.96, 0.59 against 0.91), so on the readability axis they fall in line with the L=6 ladder rather than below it.

The twins' residual. We report the residual rather than treating the two twins as an exact tie. Both have an anisotropic spectrum that the orthogonal placebo lacks, so the aggregator's first layer W_{proj} must learn the inverse spectrum. That is an optimization cost, not a representational limitation, and Proposition 1(b) does not rule it out. With the current budget we cannot separate that effect from any remaining difference between the twins. This caveat does not apply to the dose effect itself: at the highest doses, its relative size is within three percentage points of its value from 100k onward (Appendix E.3), rather than showing the delayed recovery expected from a merely slower arm.

MOF-Reach: a control that must itself be validated. On MOF-Reach, the cov-matched twin is *not a valid control*. Matching covariance conditioning requires inverting the token correlation matrix, which is nearly singular because most appearance coordinates vary little in this fixed scene. The resulting map is $14\times$ harder on $\mathrm{cond}(J)$ than the coupling arm it is intended to control, so it is a lossy arm rather than a dose-zero control and underperforms accordingly. The jac-matched twin does not invert that matrix, is matched to within $1.2\times$, and ties COMB. Because the two twins behave differently across suites, we treat validation of the control itself as part of the analysis.

E.3 Sample Efficiency: Area Under the Learning Curve

Prediction P2 implies that a larger dose should reduce return at a fixed interaction budget, which would appear as lower sample efficiency. We cannot use a threshold-crossing statistic ("steps to reach $x\%$ of the reference") because no arm has plateaued by 250k and the high-dose arms never reach several reference levels based on COMB. The statistic is therefore censored precisely where the effect is largest. Instead, we report normalized area under the evaluation curve, $\mathrm{AULC} = \frac{1}{T}\int_0^T R(t)\,dt$ with $T = 250$k, computed per seed and averaged, which is well defined for every arm and requires no convergence assumption (Table 10, OCRL-OOO-Hard and MOF-Reach). AULC orders the dosed arms as final return does on both suites. On the position $\times$ colour suites of Section 7 it does not separate dose from a slow start: GA-Cube's jac-matched twin trails COMB by 49% at 150k and by 5% at 300k, so its area is as low as that of the weakest dosed arm, and on OCRL-Reach the placebo's area falls below the weakest dosed arm's for the same reason. We therefore compare those suites at the end of a fixed budget only.

Mixer	dose η	AULC	vs. COMB	gap @100k	gap @250k
OCRL-OOO-Hard (2D, sparse reward)					
COMB (identity)	0.000	33.9 ± 1.5	–	–	–
placebo (orthogonal)	0.000	34.7 ± 1.4	$+3\%$	-6%	$+6\%$
linear twin, cov-matched	0.000	30.6 ± 1.6	-10%	-18%	0%
linear twin, jac-matched	0.000	30.6 ± 1.1	-10%	-10%	-5%
coupling L=4, λ=1.0	0.131	29.5 ± 1.0	-13%	-16%	-9%
coupling L=6, λ=0.75	0.132	30.4 ± 1.1	-10%	-16%	$+2\%$
coupling L=6, λ=0.85	0.173	26.7 ± 1.7	-21%	-30%	-17%
coupling L=6, λ=0.925	0.203	24.2 ± 1.0	-29%	-28%	-24%
coupling L=4, λ=1.38	0.222	22.8 ± 0.7	-33%	-32%	-36%
coupling L=6, λ=1.0	0.232	23.3 ± 1.2	-31%	-36%	-28%
coupling L=6, λ=1.2	0.374	18.4 ± 1.5	-46%	-38%	-48%
coupling L=6, λ=1.5	0.656	15.0 ± 0.7	-56%	-57%	-60%
MOF-Reach (3D, dense reward)					
COMB (identity)	0.000	57.5 ± 4.5	–	–	–
placebo (orthogonal)	0.000	59.8 ± 3.4	$+4\%$	$+11\%$	-2%
linear twin, jac-matched	0.000	55.5 ± 3.6	-3%	$+3\%$	-2%
coupling L=6, λ=0.75	0.096	44.3 ± 3.7	-23%	-15%	-21%
coupling L=6, λ=1.0	0.174	38.8 ± 4.2	-33%	-29%	-26%
coupling L=6, λ=1.5	0.416	24.4 ± 5.0	-58%	-52%	-52%

Table 10: Normalized area under the evaluation curve up to 250k (mean return over training), mean $\pm$ s.e.m. over seeds, and the relative return gap to COMB at 100k and at 250k, each over a trailing 25k-step window.

Why a fixed budget is not a hidden caveat. The OCRL-OOO-Hard and MOF-Reach ladders are reported at 250k without claiming convergence, and the reason this is safe is that the largest deficits are stable in *relative* terms well before the budget ends: at η=0.66 on OCRL-OOO-Hard the gap to COMB is -57% at 100k against -60% at 250k, and at η=0.42 on MOF-Reach it is -52% at both. At intermediate doses the gap moves by up to ten points between 100k and 250k, in both directions (-36% to -28% at η=0.23, -38% to -48% at 0.37), with no systematic narrowing. Neither pattern is the signature of arms that are merely slower and would eventually catch up. The dose-zero residual among the controls is a different matter and we do not make the same claim about it: the cov-matched twin trails COMB by 18% at 100k and ties it at 250k, so there a rate difference and an asymptotic one are not separable at this budget.

F Reward-Free Control from Written Objectives

Section 7 asks what a *learned* head can extract from the representation, and readability itself is measured with another learned map. Addressability is stricter: it asks whether we can specify the

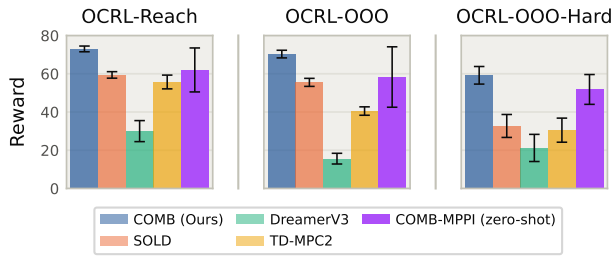


Figure 7: **Reward-free planning on a frozen COMB.** Return on the three OCRL tasks of a frozen COMB driven through a latent MPPI loop by objectives written over the named coordinates of Eq. 3 (COMB-MPPI, zero-shot), against the trained COMB policy and the trained baselines. No learned reward, goal image or retraining is used. Objectives and optimization in Appendix F.1.

task directly in the latent coordinates. If the coordinates really carry their named attributes, an objective can be written in terms of $\boldsymbol{z}_p$ and $\boldsymbol{z}_f$ without learning another head. We test this by driving a *frozen* COMB with latent MPPI, using no learned reward, goal image, or retraining (Appendix F.1).

On the three OCRL tasks, the frozen model recovers most of the return of the fully trained COMB policy (Figure 7). The same model can also reach a named object, avoid another, and visit two objects in sequence by changing only the written objective (Figure 8). A holistic latent does not expose coordinates to which such an objective can be attached. Standard alternatives specify a goal image (Mendonca et al., 2021; Zhou et al., 2025) or a prompt grounded by a foundation model (Mazzaglia et al., 2024); we do not evaluate those alternatives here. The distinction is that a goal image specifies one *instance* of a task, whereas a written objective can specify the task *itself*. Relational or negated tasks, such as "the odd one out" or "reach this and avoid that", have no fixed goal image because the target depends on the episode. SOLD provides the closest object-centric contrast: it has entities, but its slots are unstructured embeddings, so position or colour cannot be addressed without first learning a probe.

Relation to other planners. Goal-conditioned agents such as LEXA (Mendonca et al., 2021) can only be given goals that look like their own observations. Object-centric planners that impose explicit spatial constraints (Wang et al., 2021; Ferraro et al., 2024), like object-centric world models built on slots (Mosbach et al., 2025), on external segmentation (Rubinstein et al., 2025; Zhang et al., 2025) or on externalized heads (Nishimoto & Matsubara, 2025; Feng et al., 2025), keep each entity as an unstructured slot embedding, which leaves no named coordinates to write an objective over.

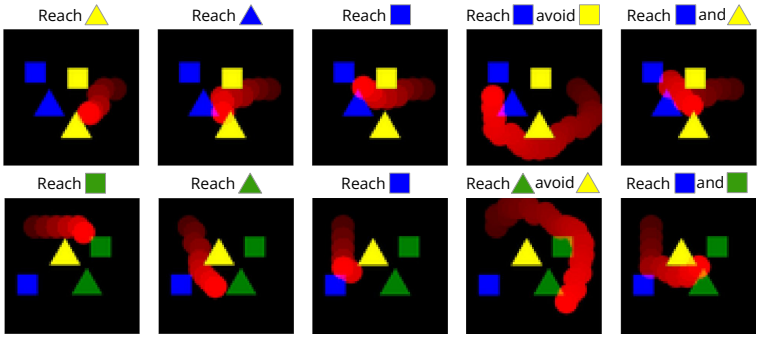


Figure 8: One frozen COMB driven by different written objectives (reach a named object, reach one and avoid another, reach two in turn), with no retraining between panels.

F.1 Objectives and Optimization

We define analytical reward functions $r(\hat{z}_t; \mathcal{G})$ directly on the predicted factorized state $\hat{z}_t = \{(z_p^{(k)}, z_d^{(k)}, z_s^{(k)}, z_v^{(k)}, z_f^{(k)})\}_{k=1}^{K}$, where $\mathcal{G}$ specifies the task constraint. The attributes can then be combined to express several task types:

- *Target Reach*: a target is identified by matching a reference appearance $z_f^\star$ against the current particles' features $z_f^{(k)}$; the objective minimizes the geometric distance between the agent and the target via their spatial coordinates $z_p^{(k)}$.
- *Odd-One-Out Reach*: pairwise cosine similarities among the appearance features $z_f^{(k)}$ of visible particles ($z_v^{(k)} > \epsilon$) select the target as the particle least similar to the cluster, and the planner minimizes the agent's spatial distance ($z_p^{(k)}$) to it.
- *Target Reach with Obstacle Avoidance*: a target and an obstacle are identified via their appearance features $z_f^{(k)}$; the reward couples an attractive term with a repulsive penalty on states where the agent approaches the obstacle's spatial coordinates.

An imagined trajectory of h steps is scored by

$$J(a_{1:h}) \;=\; \sum_{t=1}^{h} \Big[\, r(\hat{z}_t; \mathcal{G}) + w_\Delta \, \Delta r_t \,\Big], \tag{11}$$

Here $r(\cdot)$ maps latent geometry and appearance to a scalar, while Δr_t is the step-to-step change in the geometric term and serves as an optional progress bonus (w_Δ=0 disables it). No learned reward model or value head is used. We optimize Eq. 11 with Model-Predictive Path Integral (MPPI) control (Williams et al., 2017) for both continuous and discrete actions. At each iteration we sample candidate sequences from a Gaussian or categorical proposal (with Gumbel-max sampling for the latter), roll them out under the frozen action-conditioned dynamics, and reweight the better sequences to update the proposal.

G Additional Analyses

G.1 Held-Out Appearance Combinations Test the Encoder as Much as the Format

The upper block of Table 1 holds out combinations of two appearance attributes, which COMB represents in a single learned appearance code. We therefore treat these numbers as system results rather than as direct evidence for the format argument. These splits are a weaker test of format than the position × colour splits of Section 7 for two reasons. First, Proposition 1 gives no transfer shortcut when the two attributes share a code (Section 3); the head still has to extrapolate within that

			Reconstruction		Prediction	
		Method	SSIM ⇑	LPIPS ⇓	SSIM ⇑	LPIPS ⇓
OCRL OOO-Hard	ID	DreamerV3	$0.628_{\pm 0.003}$	$0.307_{\pm 0.004}$	$0.596_{\pm 0.025}$	$0.357_{\pm 0.023}$
		COMB	$\mathbf{0.894}_{\pm 0.006}$	$\mathbf{0.070}_{\pm 0.002}$	$\mathbf{0.834}_{\pm 0.020}$	$\mathbf{0.124}_{\pm 0.013}$
	OOD	DreamerV3	$0.686_{\pm 0.004}$	$0.250_{\pm 0.002}$	$0.665_{\pm 0.037}$	$0.280_{\pm 0.021}$
		COMB	$\mathbf{0.850}_{\pm 0.006}$	$\mathbf{0.119}_{\pm 0.003}$	$\mathbf{0.805}_{\pm 0.025}$	$\mathbf{0.141}_{\pm 0.017}$
GA ShapeColor	ID	DreamerV3	$0.881_{\pm 0.012}$	$0.178_{\pm 0.015}$	$0.827_{\pm 0.033}$	$0.197_{\pm 0.017}$
		COMB	$\mathbf{0.941}_{\pm 0.011}$	$\mathbf{0.092}_{\pm 0.014}$	$0.822_{\pm 0.030}$	$\mathbf{0.122}_{\pm 0.013}$
	OOD	DreamerV3	$0.859_{\pm 0.010}$	$0.215_{\pm 0.025}$	$0.784_{\pm 0.036}$	$0.242_{\pm 0.018}$
		COMB	$\mathbf{0.906}_{\pm 0.011}$	$\mathbf{0.152}_{\pm 0.020}$	$\mathbf{0.802}_{\pm 0.035}$	$\mathbf{0.179}_{\pm 0.016}$

Table 11: Reconstruction and prediction metrics for DreamerV3 and COMB, in distribution (ID) and on held-out combinations (OOD). Both world models are trained for 100k steps; predictions run 15 steps ahead. Mean over 3 seeds and 10 evaluation episodes, with confidence intervals as subscripts.

appearance vector. Second, the encoder determines what the head actually receives. In reconstructions of held-out (shape, colour) pairs, we observed cases with the correct position and shape but a colour borrowed from a pairing seen in training. A held-out episode can therefore reach the head looking like a seen case, so its return reflects encoder coverage as well as the format of the emitted code. The position $\times$ colour splits avoid both issues: the two attributes are encoded by separate mechanisms, and a held-out cell combines codes that the encoder has already seen.

G.2 Perception Check and Qualitative Results

The dose experiments assume that the frozen encoder captures the relevant scene information. On OCRL-OOO-Hard and GA-ShapeColor, COMB has better reconstruction and multi-step prediction than DreamerV3 both in distribution and on held-out combinations, with one exception: prediction SSIM for in-distribution GA-ShapeColor is similar (0.822 versus 0.827), although COMB still has lower LPIPS on that cell (Table 11; examples are shown in Figures 10 and 11). These results support using its latents for the dial and dose experiments. At the same time, they leave open how much of the policy-learning gain in Section 5 comes from perception versus the representation format. The dose experiments therefore freeze the encoder to isolate the latter.

Figure 9 compares imagined rollouts of the learned policies on held-out combinations. Each model is rolled out from the same state under the actions of its own policy: COMB's imagined rollouts keep the objects' positions and motion consistent with the real environment, while DreamerV3's blur or lose them.

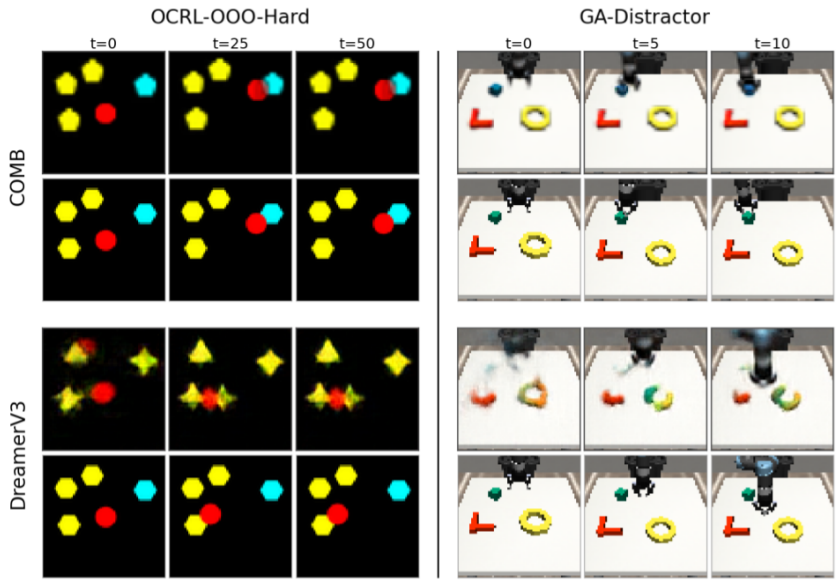
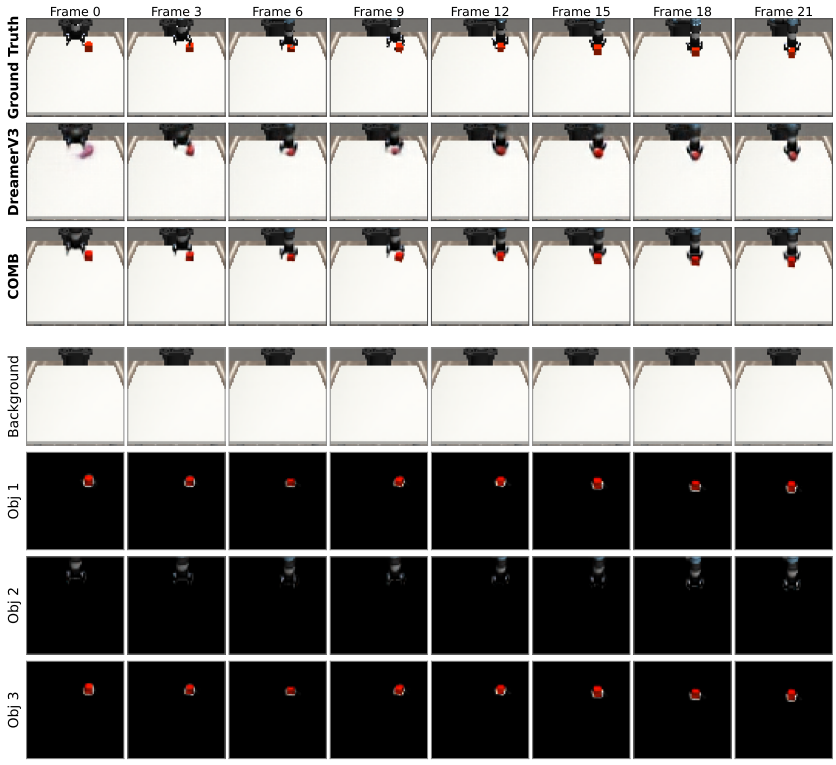
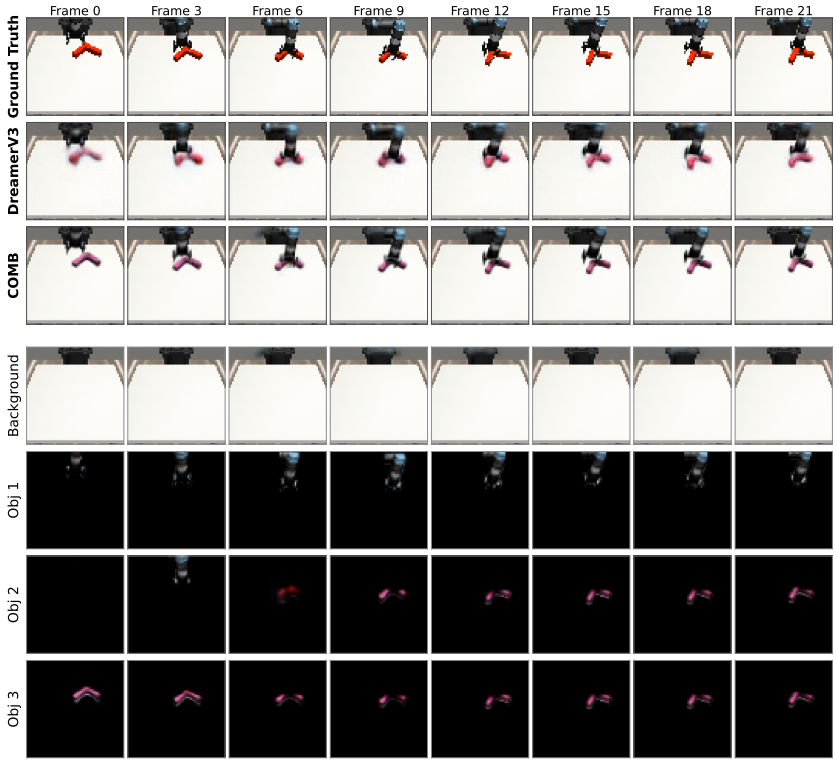


Figure 9: Qualitative rollouts on held-out (shape, colour) combinations of *OCRL-OOO-Hard* and an unseen target colour in *GA-Distractor*. Each model is initialized from the same state and rolled out autoregressively under the actions selected by its own learned policy. The second row of each block shows the same action sequence executed in the real environment for reference.

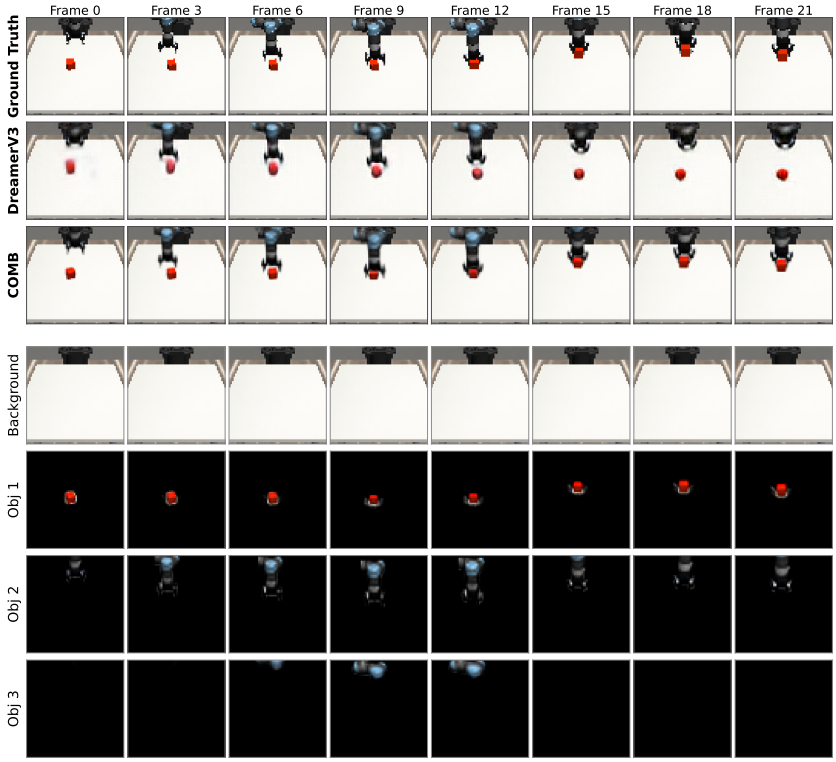


(a) GA-ShapeColor, in distribution.



(b) GA-ShapeColor, held-out combination.

Figure 10: Reconstructions by DreamerV3 and COMB, with COMB's per-particle masks below.



(a) GA-ShapeColor, in distribution.



(b) GA-ShapeColor, held-out combination.

Figure 11: Multi-step predictions by DreamerV3 and COMB, with COMB's per-particle masks below.