Stitching Manifolds: Leveraging Interaction to Compose Object Representations into Scenes.

Hamza Keurti [1 2 3] **Bernhard Schölkopf** [2] **Pau Vilimelis Aceituno** [1] **Benjamin Grewe** [1]

Abstract

In the present work, we address the problem of generalization by leveraging interaction to compose previously acquired knowledge. We show that the problem of long distance navigation can be naturally decomposed into local navigation around multiple previously known landmarks. Since these landmarks enter and exit the agent's field of view and frequently occlude each other, they must be considered collectively. We propose a two-step approach where an agent first acquires group-structured representations of individual objects by navigating around them and witnessing the changes to the view caused by its movement. In the second stage, we introduce a stitching procedure to combine the learned individual object manifolds into a coherent representation of the scene. The stitched representation is a group structured representation of the whole scene which can be maintained from any object in view and predict all other objects pose. In conclusion, the agent learns a world model representation for its navigation of the scene that is modular and data efficient, relying solely on interaction which enables it to situate itself, predict its pose evolution from performed actions and infer actions connecting two observations.

1. Introduction

A hallmark of biological intelligence is the remarkable ability to adapt and generalize rapidly to new tasks and environments by composing knowledge from various previous object related experiences. This capability is demonstrated in tasks like landmark-based navigation, where animals and human leverage spatial relationships between multiple objects to plan routes, estimate their position with respect to food sources or other animals, and acquiring new reference points.

In artificial agents, it is of significant importance to be able to decompose problems into smaller problems and employ modular computation to limit the number of parameters that need to be learned (Kirsch et al., 2018). Additionally, the possible configurations of a scene composed of a number K of objects would grow exponentially with the number of objects. We leverage compositional generalization and interaction as a shared inductive bias to learn the K single object representations separately and then efficiently 'stitch' them together, as a result the growth in complexity becomes linear.

In this work, we build artificial agents that mimic the navigational capacities of animals by forming a large-scale scene positioning system using only locally available information. Our approach is built on two key steps: First, to efficiently learn world models for navigation around individual objects and second, to merge these multiple local world models into a unified scene world model that can be used to plan and navigate coherently across multiple local reference frames.

While most multi-object representations approaches (Burgess et al., 2019; Eslami et al., 2018; Locatello et al., 2020; Kipf et al., 2020; 2022) focus on segmenting the image into its object components, we instead shift the focus to composing navigable object representations into a navigable scene representation. Each object world model is a group structured representation learned from interaction as part of a Homomorphism AutoEncoder (Keurti et al., 2023). In a given multi object scene, we use interaction cues to superpose (stitch) the learn object manifolds into a coherent group structured representation for the scene that is robust to occlusion and objects exiting the frame. We discuss related works in Appendix B.

We summarize our contributions as follow:

- We formalize the existing geometric structure of static scenes with regard to agent and object poses and movements of the agent.

[*]Equal contribution [1]Institute of Neuroinformatics, ETH Zürich, Switzerland [2]Max Planck Institute for Intelligent Systems, Tübingen, Germany [3]Max Planck ETH Center for Learning Systems. Correspondence to: Hamza Keurti <hamza.keurti@tuebingen.mpg.de>.

Proceedings of the Geometry-grounded Representation Learning and Generative Modeling at 41st International Conference on Machine Learning, Vienna, Austria. PMLR Vol Number, 2024.

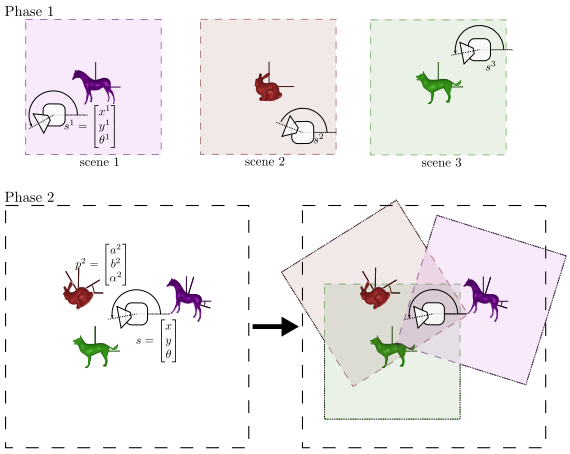

Figure 1. The agent reuses learned representations in Phase 1 from single object scenes (scene k) in scenes where multiple objects are present in phase 2.

- We propose a novel stitching approach to compose group structured representation of objects into a coherent scene representation.
- We show through experiments the procedure produces a navigable world model where the agent can predict the effect of its actions in the form of rollouts, can reason about its relative pose to each object at all times and navigate between source and target views despite occlusions and out of frame objects.

2. Learning Extended World Models for large-scale Navigation

2.1. Problem setup

In our setup, scenes are composed of multiple objects and an agent moves around and observes the scene. When the scene contains only a single object k put at a canonical pose, we call the scene single object scene k or scene k. We are interested in a two-phases setup illustrated in Figure 1. In the first phase, the agent moves around each object k separately in scene k to learn representations. In the second phase, the agent moves in a composed scene with multiple of the already encountered objects. As it moves in the composed scene, its relative pose to each object varies. We formalize this problem in sections 2.1.1 and 2.1.2. The diagram in Figure 4 summarizes all the relations between the components of this project. We invite the reader to refer to the diagram to better visualize the formalism.

2.1.1. Geometric Structure of Latent Scene

In a static scene, there exists a natural structure relating the different objects poses and the pose of a moving agent. For a visual account of this structure, the reader can refer to the 'Latent' part of the diagram Figure 4 in Appendix F.1. We consider an agent with pose $s \in S$ navigating a scene containing K objects with poses $(p^k)_{k\in[K]} \in S$. The state space for the scene is therefore $S \times \underbrace{S \times \cdots \times S}_{K}$. The pose space S of the agent and of each of the objects in the scene is a manifold structured by the regular action of a smooth group G, we can therefore identify S with G (more in Appendix A). If for instance, the pose space $S = \mathbb{R}$ is simply an object's position on a $1D$ line, then the group acting on it is $(G, \cdot) = (\mathbb{R}, +)$ and its action on S is also $+ : (g, s) \mapsto g + s$. The agent performs actions from G or a subset of it to navigate around the scene.

In particular, if the scene contains only the object k put at the origin of the scene (point of S identified with the identity of G), we will call this scene "scene k" and we denote in this scene the absolute pose s of the agent by s^k. The pose s^k also corresponds to the relative pose of the agent with regard to object k in any scene. **Notation:** In s_t^k the exponent k corresponds to the object index and the index t corresponds to the sample or the time index.

In any given scene, the relative pose s_t^k of the agent to a given object k is given by $s_t^k = (p^k)^{-1} s_t$. In the $1D$ example: $s^k = s - p^k$. This relative pose s_t^k also describes a state on S. It will be useful to note that the relative pose of an object k can be obtained from that of an object l through $s_t^k = (p^k)^{-1} p^l s_t^l$. As the agent performs a movement g_t in the composed scene, changing its state from s_t to $s_{t+1} = g_t s_t$, it induces the relative movement $g_t^k = (p^k)^{-1} g_t p^k$ with regard to object k , changing its relative pose to it from s_t^k to $g_t^k s_t^k$.

2.1.2. Interaction Probes the Latent Structure

While the latent states are hidden, we consider the agent is equipped with a measurement device (camera) to collect observations, and actions which change its state. We encourage the reader to view the 'Observation' part of the diagram in Firgure 4. An observation mechanism $b : S \times \overbrace{S \times \cdots \times S}^{K} \to O$ maps the agent's latent pose $s \in S$ and the configuration $(p^k)_{k\in[K]}$ of the K objects in the scene to the observations o, which are images in our setup. For a given scene, the poses p^k are constant, therefore we will write $o_t = b(s_t)$ instead of $o_t = b(s_t, p^1, \ldots, p^K)$. We suppose observations o_t can be segmented into the separate objects segments: $seg : o_t \mapsto (o_t^k)_{k\in[K]}$ such that $o_t = \sum_k o_t^k$. The observation segment o_t^k of object k at pose p^k with the agent at pose s_t can be obtained from scene k where the isolated object k is at the "canonical" pose and the agent at the pose $s_t^k = (p^k)^{-1} s_t$. We also consider the associated observation mechanism for scene k, $b^k : s^k \mapsto o^k$.

At any time, the agent performs actions g_t changing its pose from s_t to $s_{t+1} = g_t s_t$, and observes the tuple (o_t, g_t, o_{t+1}), which can be segmented as $\big((o_t^k)_k, g_t, (o_{t+1}^k)_k\big)$. The agent does not have access to its relative movement g_t^k with regard to each object k and therefore will need to learn it by understanding the relative pose of objects with regard to the scene. Figure 4 (Observations) summarizes these relations.

2.1.3. Desiderata of the learned representation

The core idea to represent the agent's pose and movements in a given composed scene comprising of individual objects, is to rely on the previously learned representations with regard to individual objects. Each object in the global scene is only seen from a subregion of the poses of the agent, due to occlusions or exiting the field of view. This makes it impossible to rely on a single object to describe the pose of the agent within the global scene, instead the agent will rely on one object at each time to estimate its pose in the scene. To solve this problem we developed a stitching procedure described in Section 2.2 which combines local object manifolds into one global manifold the agent can use to keep track of the objects' locations, predict trajectories and navigate the scene.

2.2. Stitching procedure composes scenes from objects

During the first stage, the agent learns vector representations $h^k : O \to Z$ for each individual object k allowing it navigating single object scenes. In a second stage, in the composed, global scene, the agent then learns matrices which link the individual representations h^k to each other enabling the prediction of the representation of the relative pose with regard to any object l from any other object l. It also learns matrices which conjugate the actions of the agent in the scene to convert them into relative action with regard to the single objects. This section details each of the component of the proposed model and procedure. We provide experimental results in Appendix E.1.

2.2.1. Single object manifold learning

We are interested in estimating the agent's relative pose to each object k and how it transforms under the agent's movements. As such the vector representation $h^k : O \to Z$ learned from the scene k should encode the relative pose of the agent to object k and satisfy equivariance under its movement. The agent only perceives sequences $(o_t^k, g_t^k, o_{t+1}^k)$ of observations and actions, this is a similar setting to the problem of learning group structured representations (Higgins et al., 2018). We therefore choose to use the Homomorphism AutoEncoder (Keurti et al., 2023) (HAE, see Appendix D.1 for more).

The k^{th} object model is an HAE (h^k, d^k, ρ), which comprises a trainable encoder $h^k : O \to Z$ mapping observations to the space of representation vectors Z, a trainable decoder $d^k : Z \to O$ reconstructing the observations from the representation space and a group representation $\rho : G \to GL(Z)$ mapping the agent's movements to invertible matrices on the vector space Z. Unlike the original HAE, we do not learn ρ but assume the agent already understands how its movements in G affect the representation space. The model learns on a set of transitions of the form $(o_t^k, g_t^k, o_{t+1}^k, g_{t+1}^k, o_{t+2}^k)$ on scene k. The k exponent in g_t^k is used to indicate the action is taken in the single object scene k, it also corresponds as previously discussed to the relative action with regard to object k in a composed scene as the object performs the action g_t. The model is trained to satisfy commutativity between action and representation, which for each transition corresponds to the equality: $h(o_{t+1}^k) = \rho(g_t^k)h(o_t^k)$. The training losses are detailed in the Appendix D.2. The associated results can be found in Appendix F.2.

As G acts regularly on S and the mapping $f^k := h^k \circ b^k : S \to Z$ is injective over the training domain, the representation manifold $h^k(O)$ in Z can be identified with G. Therefore, the object representation manifolds can be identified with each other. To evaluate how well the representation h^k is group structured, meaning it satisfies the equivariance $h^k(g \cdot o) = \rho(g)h^k(o), \forall o, g$, we use the manifold score described in Appendix D.4.

2.2.2. Composed Scene Representations

In a scene composed of K objects at poses p^k, we can reuse the previously learned mappings h^k, d^k by segmenting the observations o_t to each object's segment o_t^k. At a given pose of the agent s in the scene, we obtain the representations $h^k(o_t^k) = z_k^t$ for the subset of objects k currently in view. The representations correspond to the pose $s^k = (p^k)^{-1}s$ in the isolated object scene. And the movement g on the composed scene relates to the movement g^k on the scene k through the conjugation $g^k \mapsto p^k g (p^k)^{-1}$.

2.2.3. (Static) Stitching of objects manifolds

The relative pose s^k to object k can be expressed from the relative pose s^l with regard to object l through $s^k = (p^k)^{-1}p^l s^l$. Through equivariance we would obtain by identifying the representation manifolds with the group manifolds that $z^{G,k} = \rho((p^k)^{-1}p^l)z^{G,l}$. The encoders $(h^l)_{l\in[K]}$ map to a manifold identified with $\rho(G)$ however they are not constrained to embed it at the same location of Z. Let M^{kl} be the matrix offsetting the manifold l into the manifold k. In other words for a same pose $s_0 \in S$ in each of scenes l and k, the representations satisfy $h^k(b^k(s_0)) = M^{kl}h^l(b^l(s_0))$. Putting it together, $z^k = \rho((p^k)^{-1}p^l)M^{kl}z^l$. In the end, for a given scene, we need to estimate a constant matrix R^{kl} which satisfies for

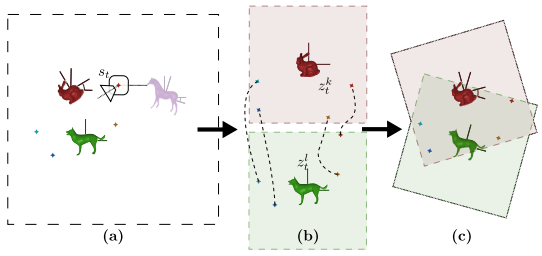

Figure 2. Static Stitching: **(a)** Collect observations where k and l are visible. **(b)** Represent in each object's manifold. **(c)** Infer R^{kl} which transforms a representation from manifold k to l.

any pose s_t of the agent: $z_t^k = R^{kl} z_t^l$ and therefore allows us to switch from the representation of object l to that of object k. We can estimate R^{kl} from a set of "stitching points", these are poses s_t for which both objects objects l and k are in frame by solving the least squares problem:

$$R^{kl} = \underset{\substack{X \in \mathbb{R}^{D\times D} \\ |X| \neq 0}}{\operatorname{argmin}} \sum_t ||z_t^k - X z_t^l||_2^2. \quad (1)$$

2.2.4. Learning a Global World Model Manifold: Stitching of object manifolds for non adjacent objects

If no or not enough observations o_t can be collected where both objects l and k are in frame, then R^{kl} cannot be estimated through the least squares problem Equation 1. It can instead be obtained through an intermediate object j adjacent to both through $R^{kl} = R^{kj} R^{jl}$. This operation can be repeated to estimate R^{kl} for objects l, k that are connected at a higher degree of separation. The path of intermediate objects $i_1, \ldots, i_I$ can be chosen on the basis of the least sum of errors achieved on the associated MSE problems Equation 1.

2.2.5. (Dynamic) Stitching of global movement to relative movement

The agent represents its pose s in the scene through representations z_t^k of its relative pose s_t^k to different objects k in the scene but lacks knowledge of the transformations g_t^k of the object representations and only observes its action g performed relative to the scene. Therefore, to predict the effect of its movement g_t on its pose, the agent needs to infer the associated relative movements g_t^k. We have established in Section 2.1.1 that the relative movement g^k of the agent is related to its absolute movement g through $g_t^k = (p^k)^{-1} g_t p^k$. As the group representation is a homomorphism we get $\rho(g_t^k) = \rho(p^k)^{-1} \rho(g_t) \rho(p^k)$. However p^k, the object's pose in the scene is unknown. The agent only needs to estimate the conjugation matrix $P^k := \rho(p^k)$.

As the agent performs an action g_t, it observes the transition (o_t, g_t, o_{t+1}), from which it can get the representations transition $(z_t^k, \rho(g_t), z_{t+1}^k)$. The transition satisfies $z_{t+1}^k = (P^k)^{-1} \rho(g_t) P^k z_t^k$. By having the agent navigate the scene and observing how its representation of object k changes, we can estimate P^k from a dataset of tuples $(z_t^k, g_t, z_{t+1}^k)_t$ through minimizing the least squares in Equation 2. While the solution to this problem is not unique, it captures the part of the reference change that matters to obtain the right transformation $\rho(g_t^k)$. Finding P^k for a single object k is sufficient as the other P^l's can be determined from the R^{kl}'s through: $P^l = R^{kl} P^k$. As optimizing over invertible matrices can be challenging we use the exponential matrix parametrization (See Appendix E.1).

$$P^k = \underset{\substack{X \in \mathbb{R}^{D\times D} \\ |X| \neq 0}}{\operatorname{argmin}} \sum_t ||z_{t+1}^k - X^{-1} \rho(g_t) X z_t^k||_2^2. \quad (2)$$

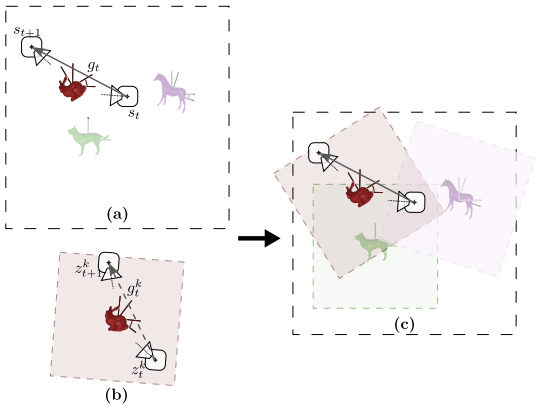

Figure 3. Dynamic stitching: **(a)** Act and observe in the global scene. **(b)** Represent on the object manifold, the relative movement g_t^k is unknown. **(c)** Infer P^k that transforms the observed g_t to g_t^k.

3. Experiments and Results

We refer to the Appendix F for details on our experimental setup (Appendix F.1) and for different results (Appendices F.2 and F.3).

3.1. Single Object Representations

The single object representations for multiple 3D objects are obtained by training an HAE model on a dataset of collected transitions. The learned representation is described in Appendix F.2.

3.2. Stitched Scene Representation

The scene representation resulting from the stitching procedure is described in Appendix F.3. We show in Appendix F.3.1 that the stitching can be used to predict, at a given time, the representation of objects from a chosen visible object. We also show in Appendix F.3.2 that the resulting scene representation can be used to predict rollouts,

where the agent views the image associated with its initial pose then performs a sequence of actions without accessing images. By virtue of the equivariance of the composing object representations, the stitching matrices, the agent can maintain the representations of all objects as they transform under its action.

4. Conclusion

We have studied the geometric and interactive structure underlying a static scene with a moving agent, and object positions and poses. We leveraged equivariant object models to compose a scene representation from the individual object representations. The resulting 'stitched' representation supports inferring the scene configuration from seeing a single object. It also supports predicting the effect of rollouts on the agent's pose relative to the objects. In particular we showed that this approach is useful for building a scene pose from relative poses to objects in a large scene. We also showed that the agent can use the resulting scene pose to navigate between a source and target observation.

Limitations In this work we focused on the composition of object representations but we did not give an account on how to segment observations into object components. Another limitation is that the objects in the scene should not present symmetries with regard to the movement of the agent as this would lead the object model to collapse. A probabilistic account would solve this by having a controller choose which objects to rely on as landmarks based on how informative on the relative pose to the agent they are.

Acknowledgements

H.K. was supported by CLS. This work was supported by the Swiss National Science Foundation (B.F.G. CRSII5-173721 and 315230_189251), ETH funding (B.F.G. ETH-20 19-01), the Human Frontiers Science Program (RGY0072/2019) and the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 1IS18039B.

References

Philipp Bader, Sergio Blanes, and Fernando Casas. Computing the matrix exponential with an optimized taylor polynomial approximation. *Mathematics 2019, Vol. 7, Page 1174*, 7:1174, 12 2019. ISSN 2227-7390. doi: 10.3390/MATH7121174.

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.

Hugo Caselles-Dupré, Michael Garcia-Ortiz, and David Filliat. Symmetry-based disentangled representation learning requires interaction with environments. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. doi: 10.1126/science.aar6170. URL https://www.science.org/doi/abs/10.1126/science.aar6170.

Alex Foo, Wynne Hsu, and Mong-Li Lee. Multi-object representation learning via feature connectivity and object-centric regularization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=BDno5qWEFh.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Hamza Keurti, Hsiao-Ru Pan, Michel Besserve, Benjamin F Grewe, and Bernhard Schölkopf. Homomorphism AutoEncoder – learning group structured representations from observed transitions. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 16190–16215. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/keurti23a.html.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1gax6VtDB.

Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional Object-Centric Learning from Video. In *International Conference on Learning Representations (ICLR)*, 2022.

Louis Kirsch, Julius Kunze, and David Barber. Modular networks: Learning to decompose neural computation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/310ce61c90f3a46e340ee8257bc70e93-Paper.pdf.

Anson Lei, Bernhard Schölkopf, and Ingmar Posner. Variational causal dynamics: Discovering modular world models from interventions. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=V9tQKYYNK1.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8511df98c02ab60aea1b2356c013bc0f-Paper.pdf.

Loek Tonnaer, Luis Armando Perez Rey, Vlado Menkovski, Mike Holenderski, and Jim Portegies. Quantifying and learning linear symmetry-based disentanglement. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 21584–21608. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/tonnaer22a.html.

Elise van der Pol, Daniel Worrall, Herke van Hoof, Frans Oliehoek, and Max Welling. Mdp homomorphic networks: Group symmetries in reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4199–4210. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/2be5f9c2e3620eb73c2972d7552b6cb5-Paper.pdf.

Tao Yang, Xuanchi Ren, Yuwang Wang, Wenjun Zeng, and Nanning Zheng. Towards building a group-based unsupervised representation disentanglement framework. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=YgPqNctmyd.

A. Background on group theory

This appendix provides the necessary concepts from group theory used in the paper.

Definition A.1 (Group)**.** A set G is a group if it is equipped with a binary operation $\cdot : G \times G \to G$ and if the group axioms are satisfied

1. Associativity: $\forall a, b, c \in G$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
2. Identity: There exists $e \in G$ such that $\forall a \in G$, $a \cdot e = e \cdot a = a$.
3. Inverse: $\forall a \in G$, there exists $b \in G$ such that $a \cdot b = b \cdot a = e$. This inverse is denoted a^{-1}.

We are often interested in sets of transformations, which respect a group structure, but are applied to objects that are not necessarily group elements. This can be studied through group actions, which describe how groups *act* on other mathematical entities.

Definition A.2 (Group Action)**.** Given a group G and a set X, a group action is a function $\cdot_X : G \times X \to X$ such that the following conditions are satisfied.

1. Identity: If $e \in G$ is the identity element, then $e \cdot_X x = x$, $\forall x \in X$.
2. Compatibility: $\forall g, h \in G$ and $\forall x \in X$, $\quad g \cdot_X (h \cdot_X x) = (g \cdot h) \cdot_X x$

The group action $\cdot_X : G \times X \to X$ induces a group homomorphism $\rho_{\cdot_X} : G \to Sym(X)$. (where $Sym(X)$ is the group of all invertible transformations of X) through:

$$\forall (g, x) \in G \times X, \quad \rho_{\cdot_X}(g)(x) := g \cdot_X x$$

The group homomorphism property of $\rho_{\cdot_X}$ comes from the group action axioms of $\cdot_X$:

$$\rho_{\cdot_X}(id)(x) = id \cdot_X x = x \quad \text{(identity)}$$

$$= id_X(x)$$

So $\rho_{\cdot_X}(id) = id_X$. and

$$\rho_{\cdot_X}(g_1 \cdot g_2)(x) = (g_1 \cdot g_2) \cdot_X x = g_1 \cdot_X (g_2 \cdot_X x) \quad \text{(compatibility)}$$

$$= \rho_{\cdot_X}(g_1) \circ \rho_{\cdot_X}(g_2)(x)$$

Equality over all of X leads to equality of the functions: $\rho_{\cdot_X}(g_1 \cdot g_2) = \rho_{\cdot_X}(g_1) \circ \rho_{\cdot_X}(g_2)$.

In what follows, we are interested in *linear group actions* in which case the acted on space is a vector space V and the induced homomorphism ρ maps G to the group $GL(V)$ of invertible linear transformations of V. This mapping is called a group representation. Actions of this type have been studied extensively in representation theory.

Definition A.3 (Group Representation)**.** Let G be a group and V a vector space. A representation is a function $\rho : G \to GL(V)$ such that $\forall g, h \in G$, one has $\rho(g)\rho(h) = \rho(g \cdot h)$.

Note that such definition is not restricted to finite dimensional vector spaces, however we will limit our study to this case, such that representations are appropriately described by mappings from G to a space of square matrices.

B. Related works

Our scene representation approach based on composition is related to the line of research on multi-object representation learning (Eslami et al., 2018; Burgess et al., 2019; Kipf et al., 2020; Locatello et al., 2020; Kipf et al., 2022; Foo et al., 2023). While these works are mostly focused on learning object centric representations to perform object detection or extraction, our works assumes segmentation and focuses on combining the object representation for navigation. In particular, compared to works on scene understanding, we are to the best of our knowledge the first to leverage the geometric structure of the scene for navigation.

Through our stitching procedure, we aim to compose modular world models of scenes from individual world models of isolated objects. Popularized by (Ha and Schmidhuber, 2018), world models encode an agent's understanding of an environment, how it evolves and how actions impact it. Different approaches attempt to learn factorized world models in terms of the independent mechanisms (Lei et al., 2023) or similar to our work, in terms of the objects contained (Kipf et al., 2020). The classic approach to learning world models relies on the joint processing of action and state. We instead rely on algebraic considerations of equivariance between the hidden states space and the representation space. Unlike MDP Homomorphisms (van der Pol et al., 2020), we do not seek to learn an abstraction of the states and actions based on some symmetry of the problem, instead we aim for the emergence of a representation that admits the movement actions as a structuring group. This is based on the idea of group structured representations (Higgins et al., 2018), in particular we use the HAE architecture (Keurti et al., 2023) for our single object models. However none of the works in this line of research leverage the learned representations compositionally (Caselles-Dupré et al., 2019; Keurti et al., 2023; Yang et al., 2022).

Modular computation is a promising way to solve large problems. (Kirsch et al., 2018) proposes to learn a controller which selects modules based on input. For our specific problem, the module selection does not require a complex controller, it is solely based on which objects reconstruct best.

C. Overview commutative diagram

We present in Figure 4 the overview commutative diagram of the stitching procedure which highlights the different components of the model and how it builds on top of existing single object models.

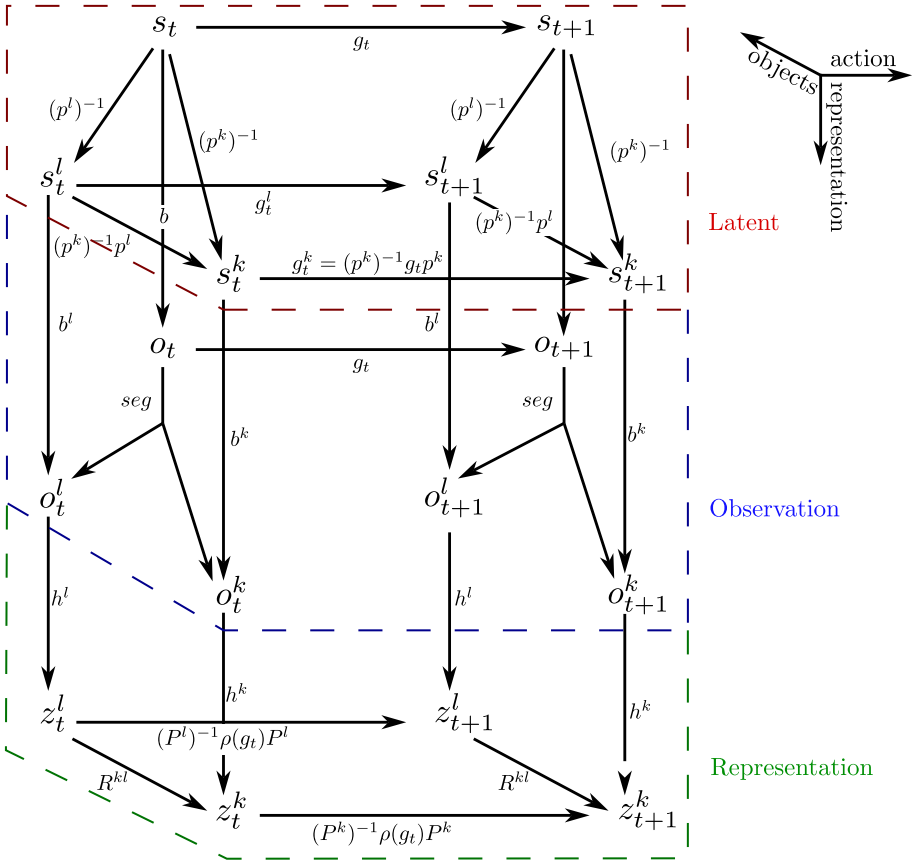

Figure 4. Absolute and relative poses and actions diagram. All of $(h^k, R^{kl}, P^k)_k$ are learned from interaction. h^k is the single object encoder, part of the object specific HAE. R^{kl} is the static stitching matrix between poses and P^k is the dynamic stitching matrix which conjugates the global action performed in the scene into relative actions performed in the object manifold.

D. Single object representation

D.1. Homomorphism AutoEncoder

The Homomorphism AutoEncoder (h, d, ρ) is an autoencoder $(h : O \to Z, d : Z \to O)$, with Z the representation space, equipped with a group representation $\rho : G \to GL(Z)$. It learns representations which satisfy the equivariance property: $h(o_{t+1}) = \rho(g_t)h(o_t)$ for any transition (o_t, g_t, o_{t+1}). In other words, acting in the real world then representing the observation yields the same representation as first representing the observation then acting in the representation space. This model learns representations constrained by the actions of the agent and learns a representation manifold similar to the pose space.

D.2. Losses

Each single object model is an HAE (Keurti et al., 2023) trained by minimizing a composite loss of a reconstruction loss (as usual autoencoders) and a latent prediction loss ensuring equivariance between the agent's actions and transformations of the representation space. Similarly to (Keurti et al., 2023), we find training on 2-step transitions to work best:

$$\mathcal{L} = \mathcal{L}_{rec} + \gamma \mathcal{L}_{pred} \tag{3}$$

$$\mathcal{L}_{pred} = \mathcal{L}^2_{pred}(\rho, h) = \sum_{t=2}^{3} \left\| h(o_t) - \Big(\prod_{i=1}^{t-1} \rho(g_i) \Big) h(o_1) \right\|_2^2 \tag{4}$$

$$\mathcal{L}_{rec} = \mathcal{L}^2_{rec}(\rho, h, d) = \sum_{t=1}^{3} \left\| o_t - d \left(\Big(\prod_{i \geq 1}^{t-1} \rho(g_i) \Big) h(o_1) \right) \right\|_2^2, \tag{5}$$

where by convention an empty product is 1.

D.3. Manifold score

In the case of a transitive action of the group G on the states space S, every state s can be reached from any chosen start state s_0 through the action of a group element g, and therefore satisfies $s = gs_0$. A group-structured representation (h, ρ) satisfies the same property in the representation space $z = \rho(g)z_0$, and it follows that $\rho(g^{-1})z$ for all representation vectors land in the same location z_0. Based on this observation (Tonnaer et al., 2022) introduces a metric to evaluate how well the learned manifold fits the structure of G by measuring the variance of the representation vectors $\rho(g^-1)z$ after applying the inverse of their associated action. With the group action being regular, we can designate an origin (canonical) state s_0 which we identify with the identity of the group and identify every other state $s = gs_0$ with the group element g. As such to evaluate how well each object manifold is learned we evaluate how $\rho(s^k_t)^{-1} z^k_t$ clusters around the representation of the canonical point:

$$m := \frac{\mathbb{V}_t[\rho(s^k_t)^{-1} f^k(s^k_t)]}{\mathbb{V}_t[f^k(s^k_t)]} = \frac{\mathbb{V}_t[\rho(s^k_t)^{-1} h^k(o^k_t)]}{\mathbb{V}_t[h^k(o^k_t)].}$$

D.4. Learned representation manifold

We show in Figure 5 the learned representation manifold for one of the objects. The representation is $4D$ augmented by a 1 to make the action of the affine translation group linear. Due to the expression of the group representation ρ, the first two components encode $(x + x_0, y + y_0)$ while the last two encode orientation through $(c\cos(\theta + \theta_0), c\sin(\theta + \theta_0))$ where c is a scaling factor and (x_0, y_0, θ_0) is an offset. We visualize the first two component through a scatter plot where we color by the true x and scale according to the true y. We visualize the orientations by estimating $\theta = arctan2(z[3], z[4])$, which we use to orient an arrow on the scatter plot. We show in Figure 6 the associated manifold score for this learned representation.

E. Stitched scene representation

E.1. Optimizing over invertible matrices

The optimization problems Equations 1 and 2 are constrained to invertible matrices. To avoid enforcing the non zero determinant constraint or computing the inverse, we use the matrix exponential to parametrize invertible matrices $X =$

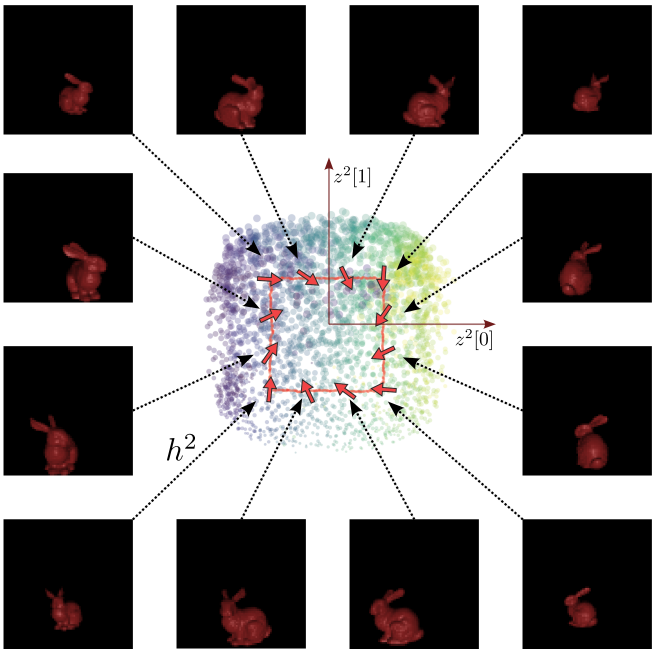

Figure 5. Learned single object representation. The scatter plot corresponds to the first two units of the learned representations $z_t^k = h^k(o_t^k)$ for a set of poses s_t^k of the agent in scene k. These units are shaped to correspond to the x (color) and y (size) of the actual pose of the agent. In red, the representations for a square trajectory of the agent around the object including few samples over the trajectory, the orientation of the arrows is obtained from the last two units of the representations, which are shaped to correspond to $(c\cos\theta, c\sin\theta)$.

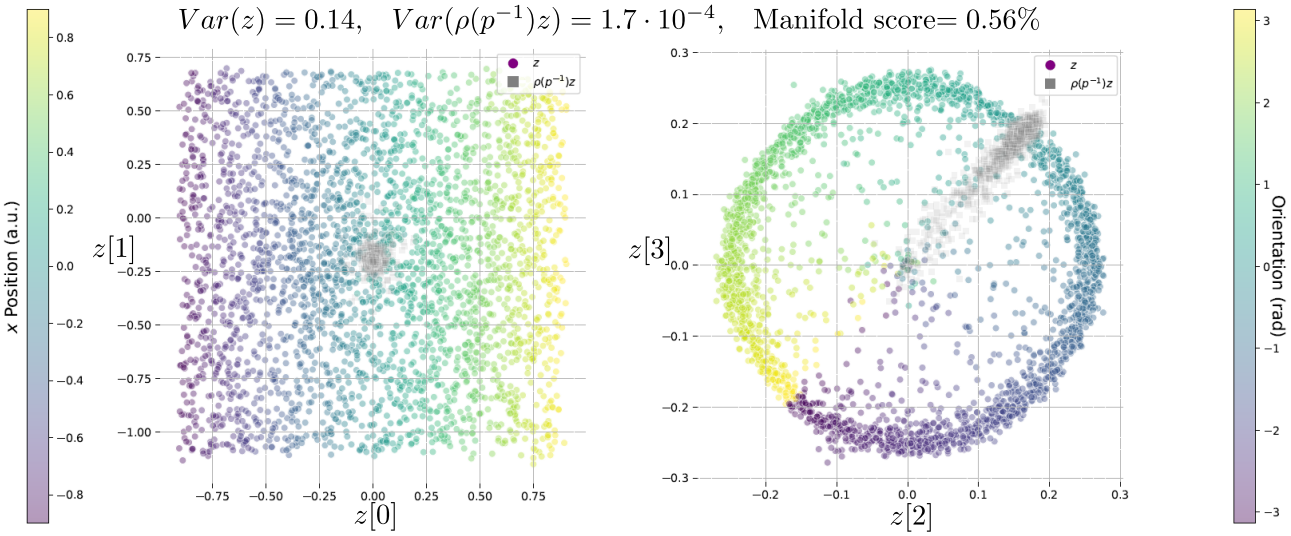

Figure 6. The manifold score for learned representation manifold of the example bunny object.

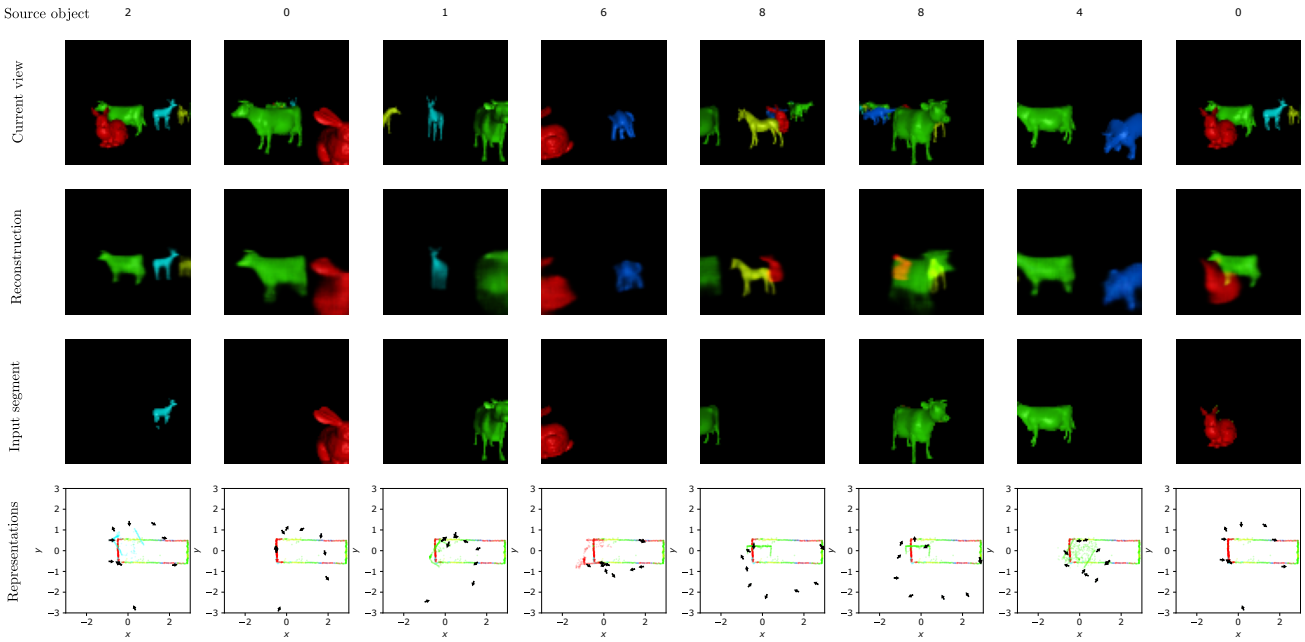

Figure 7. Representation of object 0 maintained from different source objects as the agent performs a long rectangle path around the objects. Colors indicate the source object used on that part of the path. Different arrows correspond to all object representations predicted from the reference object. This can be viewed as an animated GIF here.

$\exp(A)$ for $A \in R^{D\times D}$. (Bader et al., 2019) proposes an inexpensive differentiable approximation to the matrix exponential. The optimization problems in Equations 1 and 2 become the unconstrained problems in Equations 6 and 7.

$$R^{kl} = \exp\Big(\operatorname*{argmin}_{A\in\mathbb{R}^{D\times D}} \sum_t ||z_t^k - exp(A) z_t^l||_2^2\Big). \tag{6}$$

$$P^k = \exp\Big(\operatorname*{argmin}_{A\in\mathbb{R}^{D\times D}} \sum_t ||z_{t+1}^k - \exp(-A)\rho(g_t)\exp(A) z_t^k||_2^2\Big). \tag{7}$$

E.2. Example scene

We consider a scene containing 9 objects in a row. As such each object can only be connected to its neighbour. We learn the matrices R^{kl} for connected objects, then use composition to compute R^{kl} for disconnected objects. Note that some pairs require a product of 8 matrices to connect. We consider a single object $id = 0$ as the reference in the scene. Now wherever the agent is in the scene, it computes the representation of this one object. We show Figure 7 how the scene representation is maintained by considering different sources at each time.

F. Experimental results

In this section, we highlight the benefits of the stitching procedure on an example environment.

F.1. Experimental setup

Using Mujoco, we generate 3D scenes composed of multiple static objects. A cart like agent navigates the scene by translating in the xy plane and rotating its head, where a camera is mounted, around the vertical axis. The agent's pose $s = (x, y, \theta)$ in the scene is described by its position (x, y) and its head direction θ. Similarly, the k^{th} object pose is denoted $p^k = (a^k, b^k, \alpha^k)$, with (a^k, b^k) its position on the (xy) plane and α^k representing rotation around the vertical axis with regard to a canonical orientation. A camera on the agent's head captures image observations o_t of the scene and the image segments o_t^k for each object k. The agent can move in a given scene through displacement actions $g = ds = (dx, dy, d\theta)$.

F.2. Learned individual object models

In a first phase we train the per object models (h^k, d^k) on transitions $(o_t^k, g_t^k, o_{t+1}^k)$ from the scene k where the object k is put at the origin and at the canonical orientation $(a^k = 0, b^k = 0, \alpha^k = 0)$. With $s_t^k = (x_t^k, y_t^k, \theta_t^k)$ the pose of the agent in this scene, we limit the training domain for (x_t^k, y_t^k) to a square centered around 0 with edge L as the object is too far to be visible well otherwise, $theta_t^k$ is limited for each position to values where the object is visible. The agent's actions are given by $g_t^k = ds_t^k = (dx, dy, d\theta)$, the displacements in the scene k, which is also the change in the relative position of the agent to the object k in the coordinates of scene k. $\rho : G \to GL(Z)$ defined by the mapping

$$\rho : (dx, dy, d\theta) \mapsto \begin{bmatrix} 1 & 0 & dx & 0 & 0 \\ 0 & 1 & dy & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \cos(d\theta) & -\sin(d\theta) \\ 0 & 0 & 0 & \sin(d\theta) & \cos(d\theta) \end{bmatrix}$$

is not trained and is shared across scenes. It acts on the representations output by the encoder $h(o_t^k)$ augmented by 1 at the 3^{rd} position to use the group representation of the translation.

Each of the object specific HAEs successfully learns an encoder h^k which maps images of object k to the representation Z on a manifold structured by ρ.

Manifold: The constraint imposed by the consistency loss and the action of the group representation on the encoder h leads the representation to be similar to $(x, y, c\cos(\theta), c\sin(\theta))$ the relative pose of the agent with regard to the object up to a constant offset translation of the position (x, y) and a constant offset rotation of the orientation θ, as can be seen in Figure 5. The orientation component is learned up to a constant scaling c as only the angle matters to the rotation action. We show in Figure 6 the associated manifold score for this learned representation, as described in Appendix D.4.

F.3. Stitched scene representation

We generate a few example scenes with K ($\in [\![3, 9]\!]$) objects from the objects we trained models on. The agent renders observations o_t of the scene and navigates with actions $g_t = (dx, dy, d\theta)$ on its pose $s_t = (x_t, y_t, \theta_t)$. For each observations we also have pixel masks for each of the objects, which can be used to produce each image segment o_t^k. For each object segment o_t^k, the reconstruction error $||o_t^k - d^k(h^k(o_t^k))||_2^2$ can be used to determine if the object k is well visible. For example, a high reconstruction error could be explained by occlusion or the object being further than the training domain. We use this signal to determine if the associated representation $z_t^k = h^k(o_t^k)$ is reliable and can be used for stitching.

F.3.1. Static stitching

Learn R^{kl} : The agent randomly moves in the scene and collects a dataset of N observations $(o_t)_{t\in[N]}$. For each pair of objects (k, l), we select a subset of N^{kl} observations $(o_{t_i})_{i\in[N^{kl}]}$ where both objects k and l are visible with reliable representations. From those the agent computes the representation vectors $(z_{t_i}^k, z_{t_i}^l)_{i\in[N^{kl}]}$ from the associated segments $(o_{t_i}^k, o_{t_i}^l)$. Finally we solve the least squares problem in Equation 6 to obtain the matrices R^{kl}. We show in Figure 7 a stitched scene pose from different source objects. We also show it here in animated GIF format.

Disconnected objects : If N^{kl} is too low or zero which corresponds to objects that are never seen in the same image, we will say the two objects are not connected and rely on composition instead. After learning R^{kl} for the connected pairs of objects, we recursively compute products $R^{kl} = R^{ki}R^{il}$ until no new connection can be made. In our experiments, we only considered scenes where this process ends with all objects being connected.

Relative poses and scene pose : The benefit of the static stitching step is to provide a mean to predict an estimate of the relative pose to any object k from an object l currently in the field of view, despite the object k being occluded or out of the field of view. Indeed this provides alternative computation paths to estimate z_t^k other than $z_t^k = h^k(o_t^k)$. In the same spirit, we show that thanks to the stitching, the agent can hop from one object representation to another to maintain a continuous and coherent estimation of the relative pose representation $z_t^{k_0}$ to a single object k_0. By considering this object the origin of the scene, we obtain a representation of the pose of the agent in the scene.

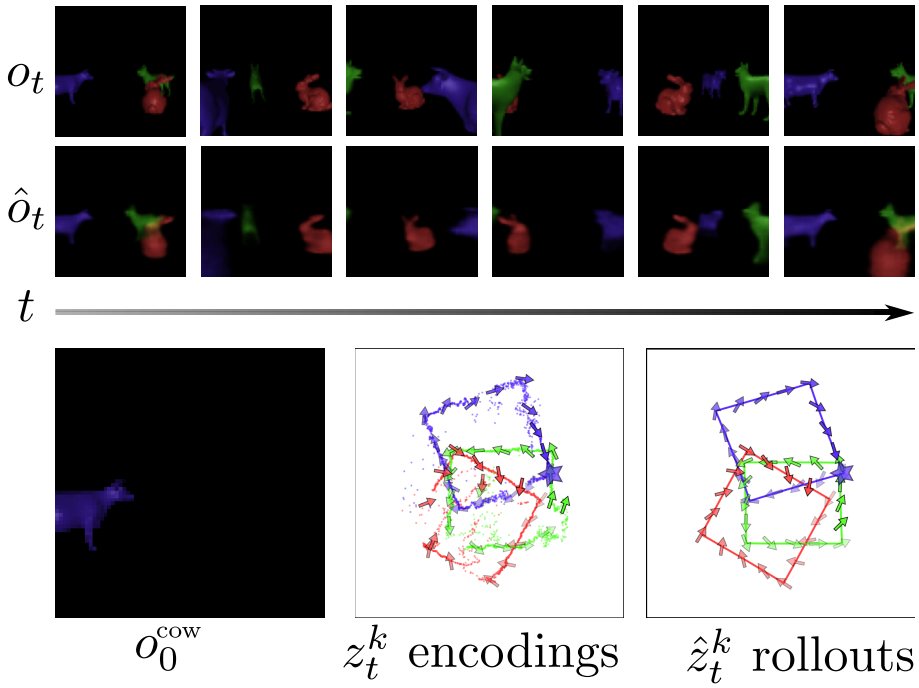

Figure 8. Rollouts along a square path from o_0^{cow} **(Bottom Left)** the cow segment of the observation o_0 at the start of the path. **Top row** shows the observations o_t along the path. **Bottom Center** shows the representations $z_t^k = h^k(o_t^k)$ for each object specific segment, the colors match the colors of the objects. The star indicates $h^{\text{cow}}(o_0^{\text{cow}})$. **Middle row** shows the reconstructions from the predicted representations $\hat{z}_t^k$.

F.3.2. Dynamic stitching

Learn P^k : The agent randomly moves in the scene and collects a dataset of N transitions $(o_t, g_t, o_{t+1})_{t\in[N]}$. For each object k, we select a subset of N^k transitions where object k is visible in both observations. These result in the representation transitions (z_t^k, g_t, z_{t+1}^k). From which P^k can be obtained from solving the problem in Equation 7.

Rollouts : With the conjugacy matrix P^k learned, the agent can transport its actions g_t to transformations $\rho(g_t^k) = (P^k)^{-1}\rho(g_t)P^k$ of the representation space of object k. As a result it can predict long rollouts in virtue of the equivariance commutative diagram its representation satisfies. From a start observation o_0, if object k is visible, then the representations after t actions can be predicted through: $z_t^k = (P^k)^{-1}\prod_{i=0}^{t-1}\rho(g_i)P^k h(o_o^k)$. In addition, through the static stitching matrices R^{kl}, the rollouts can be predicted from any start observation o_0 if a connect object l is visible, not necessarily k. Figure 8 shows rollouts for a sequence of 400 actions from a single object in o_0.

F.3.3. Source-target navigation

We highlight the benefit of our stitching procedure through a navigation task: at a given pose s_0 of the agent in the scene, the agent observes the source observation o_0. We choose a target image o_T from a random pose s_T and we ask which action g^* should the agent take to observe o_T. If o_0 contains object k and o_T contains object l, we obtain the representations z_0^k and z_T^l, which should satisfy $z_T^l = R^{lk}(P^k)^{-1}\rho(g^*)P^k z_0^k$. The action g^* can be obtained by minimizing $||z_T^l - R^{lk}(P^k)^{-1}\rho(g)P^k z_0^k||_2^2$. For source and target objects k and l that are too far apart, the errors may accumulate and g^* does not match to the minimum of this optimization problem. In this case, we can find the sequence of actions which take the agent from source to target by repeatedly solving the optimization problem and updating the source pose.