Zero-shot cross-modal transfer of Reinforcement Learning policies through a Global Workspace

Léopold Maytié
leopold.maytie@univ-tlse3.fr
CerCo, CNRS UMR5549
Artificial and Natural Intelligence Toulouse Institute
Université de Toulouse

Benjamin Devillers
benjamin.devillers@cnrs.fr
CerCo, CNRS UMR5549
Artificial and Natural Intelligence Toulouse Institute
Université de Toulouse

Alexandre Arnold
alexandre.arnold@airbus.com
Airbus AI Research

Rufin VanRullen
rufin.vanrullen@cnrs.fr
CerCo, CNRS UMR5549
Artificial and Natural Intelligence Toulouse Institute
Université de Toulouse

Abstract

Humans perceive the world through multiple senses, enabling them to create a comprehensive representation of their surroundings and to generalize information across domains. For instance, when a textual description of a scene is given, humans can mentally visualize it. In fields like robotics and Reinforcement Learning (RL), agents can also access information about the environment through multiple sensors; yet redundancy and complementarity between sensors is difficult to exploit as a source of robustness (e.g. against sensor failure) or generalization (e.g. transfer across domains). Prior research demonstrated that a robust and flexible multimodal representation can be efficiently constructed based on the cognitive science notion of a 'Global Workspace': a unique representation trained to combine information across modalities, and to broadcast its signal back to each modality. Here, we explore whether such a brain-inspired multimodal representation could be advantageous for RL agents. First, we train a 'Global Workspace' to exploit information collected about the environment via two input modalities (a visual input, or an attribute vector representing the state of the agent and/or its environment). Then, we train a RL agent policy using this frozen Global Workspace. In two distinct environments and tasks, our results reveal the model's ability to perform zero-shot cross-modal transfer between input modalities, i.e. to apply to image inputs a policy previously trained on attribute vectors (and vice-versa), without additional training or fine-tuning. Variants and ablations of the full Global Workspace (including a CLIP-like multimodal representation trained via contrastive learning) did not display the same generalization abilities.

17th European Workshop on Reinforcement Learning (EWRL 2024).

1 Introduction

Humans gather information from the world through multiple sources, leading to a rich and robust representation of their environment. Similarly, non-human agents should also learn to establish meaningful connections between information from different modalities. Such multimodal representation learning offers distinct advantages for decision-making and in particular in Reinforcement Learning. The benefits are evident when considering scenarios where one sensory input is noisy or unavailable. For instance, humans will be able to navigate in a room with subdued lighting where vision is compromised, as they can rely on other senses (hearing, touch...) to gather information about their environment. In decision-making the ability to establish links between modalities allows more efficient problem-solving, because information from one sense can be leveraged to complete or verify data from another.

For these reasons, it seems advantageous to take inspiration from human multimodal integration and apply this to embodied RL agents, e.g. for robotics. A popular theory in cognitive science about how the brain handles multimodal information is the 'Global Workspace Theory' [Baars, 1988, Dehaene et al., 1998]. According to this theory, different specialized modules compete to encode their information into a shared space called the Global Workspace. The shared representation is then broadcast back to all modules, leading to a unified interpretation of the environment. According to the theory, this last step corresponds to our inner experience. Importantly, compared to the unimodal representations in each specialized module, the shared representation enables multimodal *grounding* [Silberer and Lapata, 2012, Kiela and Clark, 2015, Pham et al., 2019], by linking objects and their properties across modalities. A deep learning-compatible adaptation of this theory has been proposed by VanRullen and Kanai [2021]. The suggested model must meet several criteria (Fig 2): an alignment of the different latent representations and the capacity to translate from one modality to the other and to broadcast signals from the Global Workspace back to each module; ideally, the model can be trained in a semi-supervised setting with unsupervised cycle-consistency objectives. An initial implementation was reported in Devillers et al. [2023], and shown to provide reliable multimodal representations that could be leveraged advantageously for downstream classification tasks, all with minimal supervision.

In this work, we explore the use of a similar multimodal representation, inspired by the Global Workspace Theory, in the context of RL tasks. In particular, we show that this model is capable of zero-shot cross-modal policy transfer, in two different environments (see section 4), each with two modalities (vision: RGB images, attributes: a vector description of the agent and its environment). The first environment is called *Factory*, a virtual factory environment simulated in Webots; the second one is called *Simple Shapes* and made of simple geometric shapes. The goal by choosing attributes and RGB images is to create two modalities that share common information without completely overlapping, particularly in *Factory* environment (see section 4). Each modality must independently contain enough information to form a policy that allows the advantages of a multimodal representation (similar to the global workspace) to be compared with those of a unimodal representation.

2 Related Work

Representation learning for Reinforcement Learning is a vast and evolving field. Sutton and Barto [1998] already discussed the importance of compact representations for an RL agent. Deep Generative models, such as Variational Autoencoders (VAEs), have the capability to encode raw data into a compact and disentangled latent space. Pioneering work by Watter et al. [2015] and Finn et al. [2016] used this approach to encode representations for Reinforcement Learning, enhancing learning efficiency from high-resolution images. Compact representations are also crucial for algorithms relying on a World Model, such as the one introduced by Ha and Schmidhuber [2018]. Further studies [Wang et al., 2023, Friede et al., 2023, Higgins et al., 2017] showed that learning disentangled environmental representations from a VAE enables agents to develop policies robust to some shifts in the original domain. Additionally, encoding observations in a well-structured space can be achieved through contrastive learning [Laskin et al., 2020, Du et al., 2021]. With this method, Gupta et al. [2017] were even able to measure policy transfer between robots having different numbers of joints.

Representation learning has now extended to multimodal RL setups. Lee et al. [2019] use fusion mechanisms with Deep Neural Networks to handle multiple sources of observations. Singh et al. [2023] align visual latent representations with graphs using a contrastive loss, while Hafner et al.

[2023] extend the work of Ha and Schmidhuber [2018] by using concatenated multimodal inputs for a world model. In a similar vein, Silva et al. [2020] extend DARLA's work [Higgins et al., 2017] to two modalities: sound and vision. They employ a multimodal VAE [Yin et al., 2017] and align representations through an additional KL loss between the two modality-specific latent spaces. This AVAE model, like ours, allows zero-shot cross-modal policy transfer, e.g. training the policy with visual inputs and using audio inputs during inference. Thus, we will use this model as a baseline to compare against our approach.

Other multimodal representation learning models like CLIP [Radford et al., 2021] have been proposed to align two (or more) latent representations, and therefore to create a common space that can be used for downstream tasks. However such models require very large amounts of paired data between modalities to learn the aligned representation in a supervised way; in a robotic context, such paired data can be difficult to obtain. In addition, it has been shown that the contrastive alignment objective of CLIP tends to discard potentially important modality-specific information [Devillers et al., 2021]. In our study, these two factors are investigated through ablation studies. First, we remove cycle-consistency objectives and train the model in a fully-supervised way. Second, we also remove the broadcast property (the ability to project global-workspace information back to each specialized module), leading to a contrastive-alignment version of our model similar to CLIP. As will be described below, both manipulations severely impair our model's ability to transfer policies between modalities.

3 Problem Formulation

Let $\mathcal{E}$ represent an environment, whose state at time t leads to an observation $o_t \in \mathcal{O}$, described as either a latent feature vector o_t^v computed from an RGB image, or an attribute vector o_t^{attr}. Based on these observations, the agent executes actions $a_t \in \mathcal{A}$ and receives a resulting reward r_{t+1}.

In this study, we first train a model to learn a representation $z_t \in \mathcal{Z}$ with two encoders $z_t^{attr} = e_{attr}(o_t^{attr})$ and $z_t^v = e_v(o_t^v)$. This step follows the approach previously described by Devillers et al. [2023], leading to a shared representation across modalities, i.e. a Global Workspace (GW). In a second step, with GW frozen, a policy π is trained to map GW-encoded observations from a specific training source $o \in \mathcal{O}^{train}$, with $train \in \{attr, v\}$, to actions $a \in \mathcal{A}$. During inference, the policy can potentially be transferred to another observation source $\mathcal{O}^{test}$, where $test \in \{attr, v\}, test \neq train$. The process is illustrated in Figure 1A, and the two training steps are further detailed below.

3.1 GW for multimodal Representation Learning

We closely follow the training setup described in Devillers et al. [2023]. That study evaluated the properties of a multimodal GW for low-resource semi-supervised training, and for downstream classification tasks; here, we are interested in applying such a system to train an RL agent. As in this previous study, we consider a setting where matched training data across modalities can be scarce or difficult to obtain, yet we have access to potentially large amounts of unimodal data (without matching labels in the other modality). Thus, we sample unimodal observations from two sets $\mathcal{U}_{attr}$ and $\mathcal{U}_v$, and paired multimodal observations from the subset $\mathcal{M} = \mathcal{U}_{attr} \cap \mathcal{U}_v$, composed of observations that are paired across both unimodal sets.

As proposed in VanRullen and Kanai [2021], Devillers et al. [2023], we do not use raw images or attributes as inputs to the GW, but encoded representations into a unimodal latent space. For images, we use a Variational Autoencoder (VAE), pretrained using the set $\mathcal{U}_v$ (see Appendix B and C for details) ; for attributes, we simply normalize them between -1 and 1. Then, we train the GW itself, composed of a set of encoders for each modality $\{e_v, e_{attr}\}$ with their corresponding decoders $\{d_v, d_{attr}\}$ (Figure 2A). The role of the encoders is to project the two unimodal latent representations onto a shared one (the GW), where they should be aligned. The role of the decoders is to allow broadcast from GW back to the unimodal representations. The training dataset $\mathcal{U}_v$ and $\mathcal{U}_{attr}$ are collected by uniformly sampling the environment in *Simple Shapes*. For *Factory* we sampled with a constraint that the table should be at least partially visible from the robot viewpoint.

To train the network, four different losses are used [Devillers et al., 2023] (see Supplementary Material for losses definitions). The translation (L_{tr}) and contrastive alignment (L_{cont}) losses are supervised losses, optimized using the set $\mathcal{M}$. The full-cycle (L_{cy}) and demi-cycle (L_{dcy}) consistency losses are optimized using the full sets $\mathcal{U}_{attr}$ and $\mathcal{U}_v$. Figure 2B illustrates how these losses are

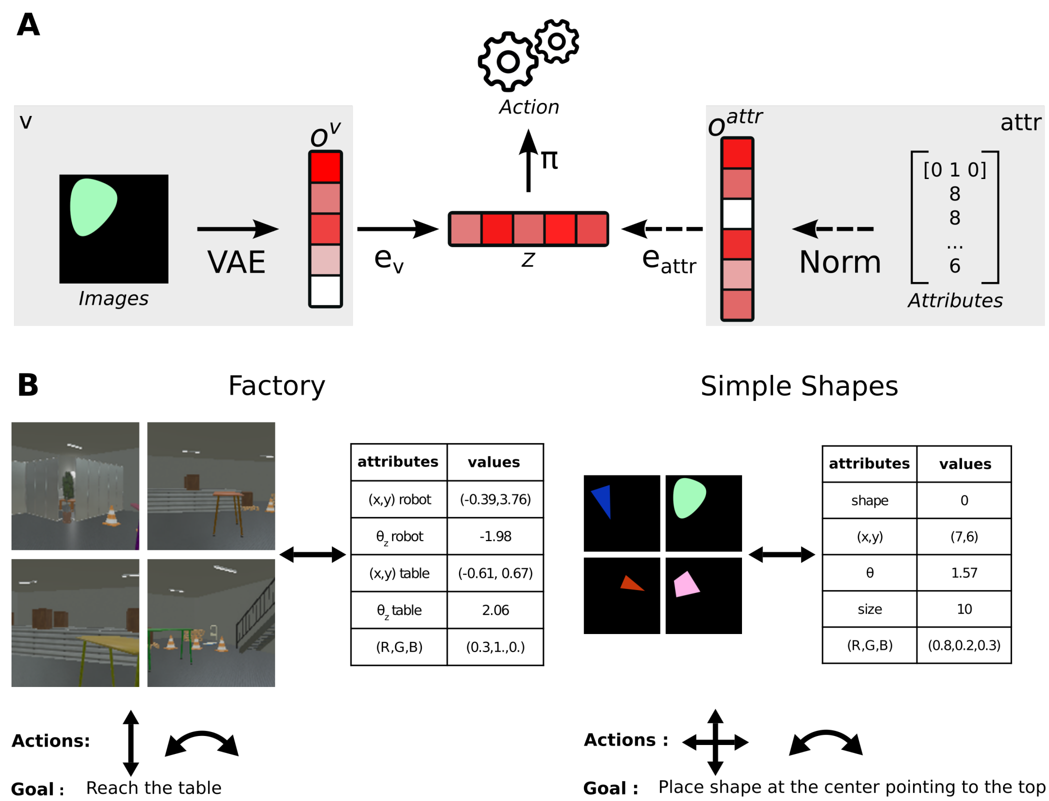

Figure 1: A: Overview of the general approach. Raw attributes are encoded in their latent representation thanks to pre-trained models (VAE arrow for images and Normalization arrow for attributes). Latent image or attribute representations can be encoded into a shared space $z \in \mathcal{Z}$ (the Global Workspace or GW) via encoders e_v and e_{attr} (respectively). The policy is trained (solid arrows) with observations from a given modality (here vision), with GW frozen. At inference time the policy can be tested with observations from a different modality (here attributes, dashed arrow); this is defined as *zero-shot cross-modal transfer*. B: Illustration of the two environments and tasks: Factory (left) and Simple Shapes (right). Example images and attributes are presented for each. For *Factory*, the agent must reach the table by rotating and moving forward or backward. For *Simple Shapes*, the agent must place the object at the center and pointing upwards, by moving to the right, left, top or down and rotating.

computed using the encoders and decoders of the GW. The total loss is a weighted sum of these four. Devillers et al. [2023] described implicit relations between the different losses, such that optimizing a subset of the losses can indirectly improve the others. By combining the four losses, the GW model optimizes the desired criteria of multimodal representation alignment and broadcast, while taking full advantage of unsupervised training data.

3.2 Policy Learning and cross-modal transfer

We use Proximal Policy Optimization (PPO), a widely adopted Reinforcement Learning algorithm introduced by Schulman et al. [2017]. We also tested Advantage Actor Critic (A2C) introduced by Mnih et al. [2016], to validate our results on another algorithm (see Supplementary Materials).These two algorithm were implemented thanks to the stable baselines 3 library [Raffin et al., 2021].

To obtain an upper baseline for cross-modal transfer, we train two policies in a more classical way using only unimodal information (the two policies' inputs are the unimodal representations of images o^v or attributes o^{attr}). This is compared with policies trained from GW-encoded representations of the observations, and tested either with observations from the same modality or from the opposite modality (i.e. zero-shot cross-modal transfer).

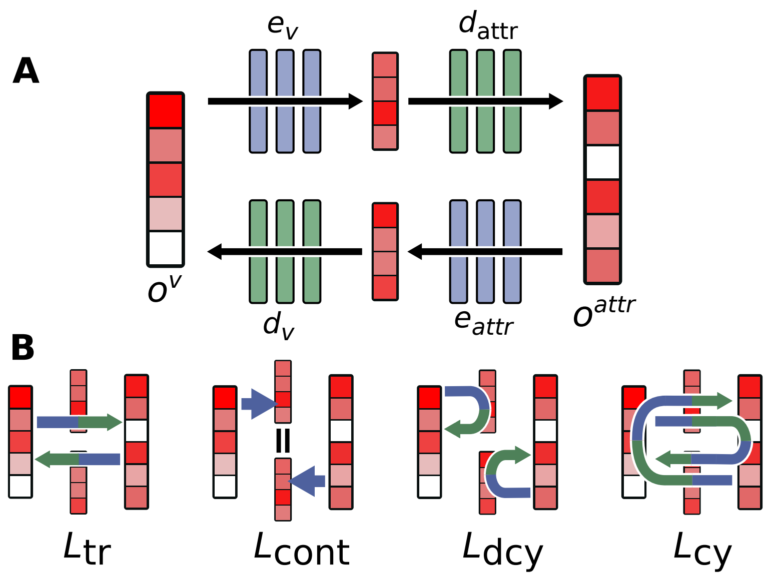

Figure 2: A: A generic view of the architecture of the Global Workspace where o^v and o^{attr} are encoded representations of the two modalities (vision, attributes). e_v, e_{attr} are feed-forward encoders into the GW representation, and d_v, d_{attr} are feed-forward decoders. Encoded representations of the two modalities $e_v(o^v)$ and $e_{attr}(o^{attr})$ are separate in the architecture, but can be aligned by virtue of the training objectives (illustrated in B), resulting in a shared GW. B: Illustration of the losses used during training of the encoders and decoders. The arrows represent the path used by the data to compute the losses. L_{tr} and L_{cont} are supervised losses for translation and contrastive alignment, respectively; they require paired training samples across the two modalities. In contrast, L_{dcy} and L_{cy} are self-supervised losses for demi-cycle and full-cycle consistency, respectively; they can be trained with unpaired samples from each modality.

While our main test relies on a GW trained using all four losses (Figure 2B), we also trained policies from GW models optimized with fewer losses, serving as ablations of the full model. A GW trained in a fully supervised way (without the cycles losses L_{cy} and L_{dcy}) serves to assess the impact of semi-supervision, especially in low-data regimes (i.e., with few paired data in $\mathcal{M}$). We also trained a policy using a GW trained only with a contrastive loss L_{cont}. This ablation evaluates the impact of "broadcast" on the performance, and serves as a CLIP-like baseline because it is trained with the same alignment objective as CLIP [Radford et al., 2021]. Finally, we compare our GW to an adaptation of the AVAE model used in Silva et al. [2020]. We modify their visual VAE to match the architecture of our own visual VAE in each environment; we also replace their audio VAE by an attribute VAE, with an architecture adapted to match the dimensions of our attribute vectors (see Supplementary Material for architecture details). This transition from audio to attribute VAE also leads to a change in the reconstruction loss: we use the same attribute reconstruction loss as the one used in the GW (see Supplementary Material). Apart from these architectural changes, the AVAE model is trained in a supervised way (on the paired multimodal set $\mathcal{M}$), as described in the original paper [Silva et al., 2020].

For both environments, we evaluate policies based on multimodal systems (GW, GW without cycles, CLIP-like, AVAE) trained with two data regimes: either a large amount of matched data (500 000 for *Simple Shapes* and 200 000 for *Factory*), $\mathcal{M} \equiv \mathcal{U}_{attr} \equiv \mathcal{U}_v$ (full data regime), or a small amount of paired data (low data regime: $\mathcal{M}$ contains 1/4th of the full dataset for *Factory*, 1/100th of the full dataset for *Simple Shapes*). This assesses the impact of the unsupervised cycle-consistency losses, and the performance of fully supervised models in a low data regime.

4 Environments

We evaluate our approach on two different environments. Each one captures observations across the same two modalities: attributes describing the state of the agent, or an RGB image. The first environment, called ‘Factory’ is a simulated factory shop floor in a robotic simulator: Webots. The second environment, named ‘Simple Shapes’ because it depicts a 2D shape on a dark background, is simulated directly using a Python-based Gymnasium environment [Towers et al., 2023].

4.1 Factory Environment

Simulated in Webots, this environment represents a factory-like shop floor with a Tiago robot and a table. The agent receives RGB images (128x128 pixels) from the robot’s viewpoint, or a set of seven attributes describing the robot and table states (Figure 1B). Robot state attributes include position (x_r, y_r) and rotation θ_r. Table state attributes include position (x_t, y_t), rotation θ_t, and color h_t. The color is defined only by the Hue of HSV, with saturation and value set to 1 to retain high-contrast colors. The final attribute state vector concatenates attribute transformations: applying cosine and sine to angles, normalizing all attributes between -1 and 1, and decomposing the table’s Hue into a cosine-sine vector.

This environment displays an asymmetry between modalities, whereby images only provide partial information while attribute vectors offer exact information, even when the robot is not facing the table. At the beginning of each episode, table attributes are randomly sampled within their domains. The robot is placed near the center with a random angle, and the agent’s goal is for the robot to reach the table. The agent directly controls the position and rotation of the robot. The robot can move forward/backward and rotate (by a maximum of 5cm and $\frac{\pi}{16}$ radians during each step). Collisions with simulation objects (e.g. walls) lead to episode termination with a penalty of -10000. At each timestep, the reward is equal to minus the distance between the robot and the table (in meters) minus $10\times$ the angle (in radians) between the robot orientation and the robot-table vector, thus penalizing the agent for not facing the table. This approach aims to guide the robot to first locate the table by rotating and then move towards it, dividing the learning into two distinct goals and enhancing performance in scenarios where the agent relies solely on the robot’s vision. When the robot reaches the table, the episode concludes with no additional reward.

4.2 Simple Shapes Environment

The second environment, called ‘Simple Shapes’, was introduced in Devillers et al. [2023]. The agent can receive two types of observations: 32×32 pixel RGB images of a 2D shape on a black background, or a set of eight attributes directly describing the environment’s state (Figure 1B). There are three different types of shapes, an egg-like shape, an isosceles triangle, and a diamond. They are represented by the variable $shape \in \{0, 1, 2\}$. The shape possesses a size $s \in [s_{min}, s_{max}]$, a position $(x, y) \in [\frac{s_{max}}{2}, 32 - \frac{s_{max}}{2}[^2$, a rotation $\theta \in [0, 2\pi[$ and an HSL color $(c_h, c_s, c_l) \in [0, 1]^2 \times [l_{min}, 1]$. The final attribute state vector concatenates transformations of these attributes: decomposing the rotation angle θ into $(c_\theta, s_\theta) = (cos(\theta), sin(\theta))$; translating HSL colors to the RGB domain, expressing the $shape$ variable as a one-hot vector of size three, and normalizing all the variables between -1 and 1.

At the beginning of each episode, attributes are randomly sampled within their respective domains. The agent’s goal is to move the shape to the center of the image at $(x, y) = (16, 16)$ and align it to point to the top, $\theta = 0$. Actions available to the agent include moving the shape by one pixel in cardinal directions (left, right, up, or down) and rotating the shape by an angle of $\frac{\pi}{32}$ clockwise or anti-clockwise. The reward is initialized at zero. At each timestep, the reward is equal to minus the current distance (in pixels) between the shape’s position and the image center minus $10\times$ the smallest angle (in radians) between the shape’s orientation and the null angle. The episode ends when the shape reaches the goal state, with no additional reward.

5 Results

The performance (average episode return) of policies trained (via the PPO algorithm) using latent representations from different models (GW and its baselines) and in different test settings is shown in Figure 3. Results for the *Factory* environment are shown in the top panels, and in the bottom panels for the *Simple Shapes* environment. In each case, models trained with a Full data regime are plotted on the left, and with a Low data regime on the right.

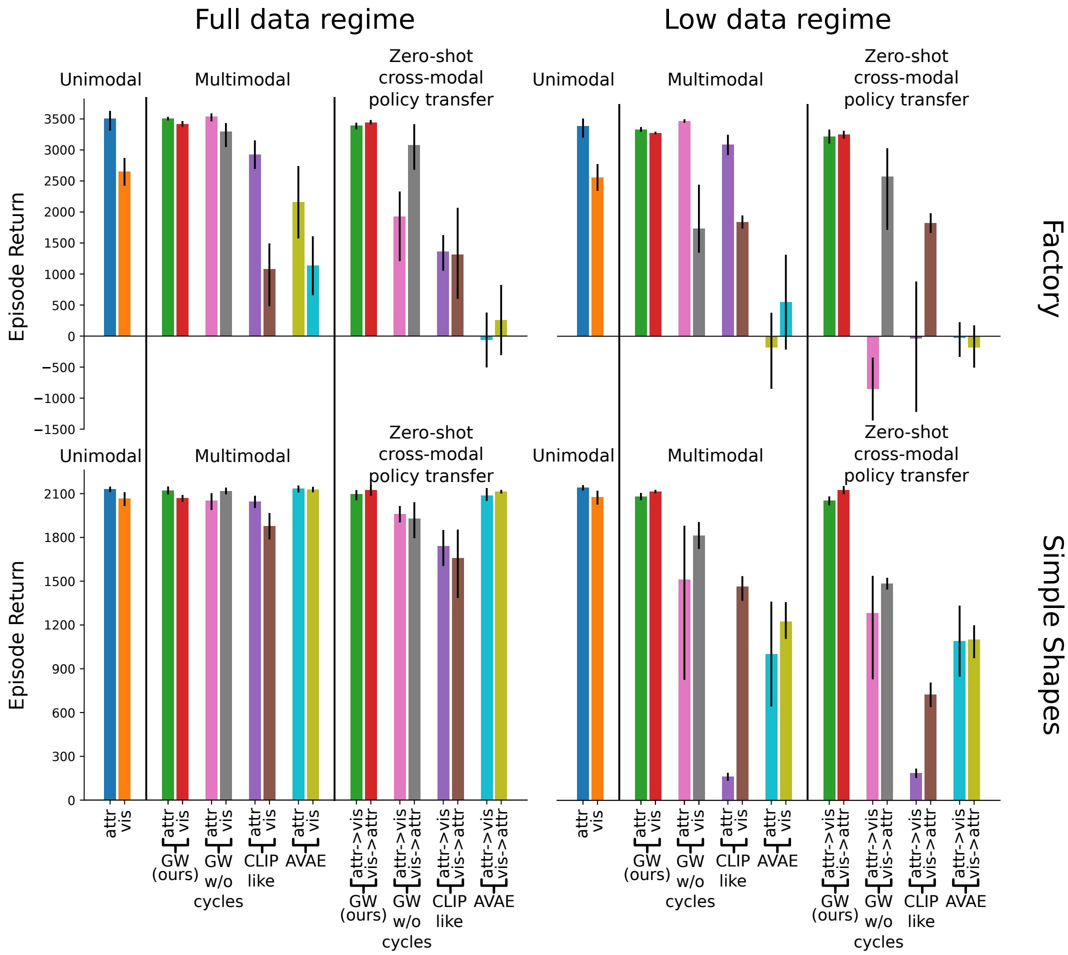

Figure 3: Performance (average episode return) of PPO trained using different latent representations and tested in different settings. A fixed value was subtracted from the episode return, corresponding to the performance of a fully-random policy in each environment; thus the random policy performance (chance level) is equal to zero in all plots (negative values reflect a defective strategy, e.g. systematically hitting walls and receiving penalties). All results are averaged across five different runs (different random seeds for policy training), and the error bars reflect 95% confidence intervals computed via bootstrapping. Models trained in the *Factory* environment are plotted in the top row, and in the bottom row for the *Simple Shapes* environment. Multimodal networks trained with all paired data are plotted in the left column (Full data regime); in the right column, the networks only have access to a subset of multimodal paired data (Low data regime). Each plot is divided into three parts: PPO trained directly from a *unimodal* latent representation; PPO trained and tested on the same *multimodal* latent representation; PPO trained on one multimodal latent representation and tested on the other (*zero-shot cross modal policy transfer*). In any given plot, bars sharing the same color depict the same trained model, tested in different settings.

We first focus on the performance of PPO trained directly on *unimodal* representations, visible on the left part of each plot in Figure 3. As expected, *unimodal* PPO acts as an upper baseline in the *Simple Shapes* environment, which is fully observable from each input modality. This is not the case in *Factory*, where PPO trained from attributes performs better than from vision; this highlights the asymmetry between visual inputs (partial observation) and attributes (entire state observation) in this environment.

The performance of PPO trained and tested on *multimodal* latent representations obtained in a Full data regime are reported in the middle part of the two left plots in Figure 3. In both environments, GW and GW without cycles yield similar rewards as the upper baseline (PPO trained directly from attributes). AVAE achieves similar performance in *Simple Shapes*, but degraded performance in *Factory*. Finally, the CLIP-like model performs poorly in both environments. We can also highlight that in *Factory*, policies trained from GW and (to some extent) GW w/o cycles are able to bridge the performance asymmetry between vision and attribute inputs. This is an example of *multimodal grounding* in the GW, whereby the learned multimodal latent representation of visual inputs is richer and more informative for a downstream decision task than the unimodal visual latent representation. The difference with results from the CLIP-like model reveals the importance of adding broadcast objectives in addition to contrastive alignment.

In the Full data regime scenario, both supervised and semi-supervised GW models (with and without cycles) perform near-optimally when trained and tested on the same *multimodal* latent representations. However, the GW cycles are particularly important when we consider the Low data regime scenario (middle part of the plots on the right in Figure 3). Here, we actually observe a drop in PPO performance for all the models in at least one input modality, except for the full GW. The decreased performance of GW w/o cycles highlights the crucial role played by the unsupervised cycle-consistency objectives in maintaining broadcast and alignment properties when the amount of multimodal paired data is low.

Finally, the zero-shot cross-modal policy transfer capabilities are shown on the right part of each plot of Figure 3. In both the full and low data regimes, and for both environments, the full GW allows for nearly optimal zero-shot transfer between modalities: a policy trained and tested on GW latent representations of attributes performs equally well when tested on GW latent representations of images (green bars), and vice-versa (red bars). The AVAE model is the only other model that permits a similar zero-shot transfer, but only in one of the four experimental settings—*Simple Shapes* in the Full data regime. In the Low data regime of *Simple Shapes* and in both regimes of *Factory*, the policy trained in one AVAE modality does not transfer well to the other. This is also the case for the CLIP-like baseline and for the GW w/o cycles ablation, in all four experimental settings.

In summary, policies learned from a GW latent representation are particularly efficient, and in some cases (e.g., *Factory*) can even surpass policies trained from unimodal representations. In addition, only policies trained from GW latents could systematically generalize to the opposite modality (zero-shot cross-modal transfer). We found that relying only on a contrastive alignment objective to establish a multimodal space (like CLIP) was insufficient. The introduction of broadcast objectives (supported by the GW decoders, see Figure 2) compels the GW encoders to retain most information present in the original unimodal latents, so that they can be accurately reconstructed by the broadcast operation. Such a GW can be trained in a purely supervised way (GW w/o cycles) when both modalities provide fully-observable information (*Simple Shapes*) and when large amounts of paired multimodal data are available for supervised training (Full data regime). In all other scenarios, the inclusion of unsupervised cycle-consistency objectives (full GW model) proves beneficial in preserving information and maintaining alignment between multimodal representations.

6 Conclusion

Our study applied a multimodal representation learning approach previously proposed by Devillers et al. [2023] (an adaptation of the Global Workspace Theory from Cognitive Science) to the training of an RL agent. The implemented model enables the construction of a multimodal latent space, allowing the encoding of unimodal information and exploiting the synergies between the different modalities. We demonstrated the capability of a GW to enable zero-shot cross-modal policy transfer, illustrating the adaptability and generalization of the learned policies across diverse modalities. Additionally, we highlighted the potential advantages of employing a semi-supervised learning framework, as seen in GW with cycle-consistency, especially in scenarios where data collection can be costly. Using a GW to generate multimodal representations, instead of other existing methods such as CLIP [Radford et al., 2021] or AVAE [Silva et al., 2020], was found to improve policy performance as well as zero-shot policy transfer across modalities. This approach not only showcases the potential of the Global Workspace Theory in enhancing the performance of RL agents, but also opens avenues for the development of more robust and versatile artificial intelligence systems capable of seamlessly transferring knowledge between different sensory domains. However, the choice of attributes as a second modality may not fully capture the complexities of real-world applications. Using a textual or proprioception (joints position of the robot) modality could present more significant challenges and provide a more realistic assessment of the system's capabilities. However, using sentences to describe agent's state in such control environment would be very similar to using attributes. For this reason, testing this approach in a more real-world environment is crucial for validating our findings and ensuring the robustness and applicability of the developed systems.

References

Bernard J. Baars. *A Cognitive Theory of Consciousness*. Cambridge University Press, New York, 1988.

Stanislas Dehaene, Michel Kerszberg, and Jean-Pierre Changeux. A neuronal model of a global workspace in effortful cognitive tasks. *Proceedings of the National Academy of Sciences*, 95(24): 14529–14534, November 1998. doi: 10.1073/pnas.95.24.14529. URL `https://www.pnas.org/doi/full/10.1073/pnas.95.24.14529`. Publisher: Proceedings of the National Academy of Sciences.

Carina Silberer and Mirella Lapata. Grounded Models of Semantic Representation. In Jun'ichi Tsujii, James Henderson, and Marius Paşca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1423–1433, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `https://aclanthology.org/D12-1130`.

Douwe Kiela and Stephen Clark. Multi- and Cross-Modal Semantics Beyond Vision: Grounding in Auditory Perception. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2461–2470, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1293. URL `https://aclanthology.org/D15-1293`.

Hai Pham, Paul Pu Liang, Thomas Manzini, Louis-Philippe Morency, and Barnabás Póczos. Found in Translation: Learning Robust Joint Representations by Cyclic Translations between Modalities. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6892–6899, July 2019. ISSN 2374-3468. doi: 10.1609/aaai.v33i01.33016892. URL `https://ojs.aaai.org/index.php/AAAI/article/view/4666`. Number: 01.

Rufin VanRullen and Ryota Kanai. Deep learning and the Global Workspace Theory. *Trends in Neurosciences*, 44(9):692–704, September 2021. ISSN 0166-2236, 1878-108X. doi: 10.1016/j.tins.2021.04.005. URL `https://www.cell.com/trends/neurosciences/abstract/S0166-2236(21)00077-1`. Publisher: Elsevier.

Benjamin Devillers, Léopold Maytié, and Rufin VanRullen. Semi-supervised Multimodal Representation Learning through a Global Workspace, June 2023. URL `http://arxiv.org/abs/2306.15711`. arXiv:2306.15711 [cs, q-bio].

Webots. http://www.cyberbotics.com. URL `http://www.cyberbotics.com`.

R.S. Sutton and A.G. Barto. Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, 9(5):1054–1054, 1998. doi: 10.1109/TNN.1998.712192.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper_files/paper/2015/hash/a1afc58c6ca9540d057299ec3016d726-Abstract.html`.

Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE, 2016.

David Ha and Jürgen Schmidhuber. World Models. March 2018. doi: 10.5281/zenodo.1207631. URL `http://arxiv.org/abs/1803.10122`. arXiv:1803.10122 [cs, stat].

Xin Wang, Hong Chen, Yuwei Zhou, Jianxin Ma, and Wenwu Zhu. Disentangled Representation Learning for Recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):408–424, January 2023. ISSN 1939-3539. doi: 10.1109/TPAMI.2022.3153112. URL `https://ieeexplore.ieee.org/abstract/document/9720218`. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

David Friede, Christian Reimers, Heiner Stuckenschmidt, and Mathias Niepert. Learning Disentangled Discrete Representations. In Danai Koutra, Claudia Plant, Manuel Gomez Rodriguez, Elena Baralis, and Francesco Bonchi, editors, *Machine Learning and Knowledge Discovery in Databases: Research Track*, Lecture Notes in Computer Science, pages 593–609, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-43421-1. doi: 10.1007/978-3-031-43421-1_35.

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 *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1480–1490, Sydney, NSW, Australia, August 2017. JMLR.org.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5639–5650. PMLR, November 2020. URL `https://proceedings.mlr.press/v119/laskin20a.html`. ISSN: 2640-3498.

Yilun Du, Chuang Gan, and Phillip Isola. Curious Representation Learning for Embodied Intelligence. pages 10408–10417, 2021. URL `https://openaccess.thecvf.com/content/ICCV2021/html/Du_Curious_Representation_Learning_for_Embodied_Intelligence_ICCV_2021_paper.html`.

Abhishek Gupta, Coline Devin, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Learning Invariant Feature Spaces to Transfer Skills with Reinforcement Learning, March 2017. URL `http://arxiv.org/abs/1703.02949`. arXiv:1703.02949 [cs].

Michelle A. Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8943–8950, May 2019. doi: 10.1109/ICRA.2019.8793485. URL `https://ieeexplore.ieee.org/document/8793485`. ISSN: 2577-087X.

Kunal Pratap Singh, Jordi Salvador, Luca Weihs, and Aniruddha Kembhavi. Scene Graph Contrastive Learning for Embodied Navigation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10850–10860, Paris, France, October 2023. IEEE. ISBN 9798350307184. doi: 10.1109/ICCV51070.2023.00999. URL `https://ieeexplore.ieee.org/document/10377327/`.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering Diverse Domains through World Models. *CoRR*, abs/2301.04104, 2023. URL `https://doi.org/10.48550/arXiv.2301.04104`.

Rui Silva, Miguel Vasco, Francisco S Melo, Ana Paiva, and Manuela Veloso. Playing Games in the Dark: An Approach for Cross-Modality Transfer in Reinforcement Learning. *Proceed ings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages pp. 1260–1268, 2020.

Hang Yin, Francisco Melo, Aude Billard, and Ana Paiva. Associate Latent Encodings in Learning from Demonstrations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), February 2017. ISSN 2374-3468. doi: 10.1609/aaai.v31i1.11040. URL `https://ojs.aaai.org/index.php/AAAI/article/view/11040`. Number: 1.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, July 2021. URL `https://proceedings.mlr.press/v139/radford21a.html`. ISSN: 2640-3498.

Benjamin Devillers, Bhavin Choksi, Romain Bielawski, and Rufin VanRullen. Does language help generalization in vision models? In Arianna Bisazza and Omri Abend, editors, *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 171–182, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.conll-1.13. URL https://aclanthology.org/2021.conll-1.13.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv:1707.06347 [cs].

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1928–1937. PMLR, June 2016. URL https://proceedings.mlr.press/v48/mniha16.html. ISSN: 1938-7228.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium, March 2023. URL https://zenodo.org/record/8127025.

EWRL Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: This paper present RL policy transfer across different modalities thanks to a brain-inspired multimodal representation. Using other models doesn't lead to such good results as the Global Workspace as claimed in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We are aware of the limitations of our model, which we discussed in the conclusion (Section 6). Using attributes instead of more complex and less overlapping modalities like text or proprioception could have lead to more realistic case alongside with less controlled environments.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results, only experimental ones.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: In Section 3 we introduced information to introduced the problem (dataset, global architecture, different step of training). This information is supplemented by more detailed one on the architectures, the losses, the rewards, etc in Appendices B, C, D and E

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While EWRL does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: For the moment only the code for the Simple Shapes environment is available on Github with checkpoints of the models used for the experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the EWRL code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the EWRL code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [No]

   Justification: Training and Test details are present in Appendices B and D but not in the core of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Figures 3 and 6 show a 95% Confidence Interval for each bar. These confidence intervals are performed with 5 different runs each time.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All information about computational resources were put in Section .

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the EWRL Code of Ethics `https://ewrl.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We don't see any violation of the EWRL Code of Ethics in this paper. To satisfy at 100% all the points we plan to release a public repository with our code.

   Guidelines:

   - The answer NA means that the authors have not reviewed the EWRL Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The results are obtained on controled environment with small dataset and modalities (like attributes) that cannot always being used in reality. To see concrete impact on society other steps of scaling in complexity and size should be perform alongside with more realistic use case

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We think that the paper doesn't present such risk. The model is fully trained internally on controlled datasets without a large pretrained model

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We took inspiration of the Global Workspace architecture from another paper as said and cited many times in Section 3. We also used PPO from stable baselines 3 that we cited in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: For the moment we didn't release any new asset.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the EWRL Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the EWRL Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

A Code availability

We provide the code for our experiments, pretrained models, and our environments here: `https://github.com/leopoldmt/RL_Simple_Shapes.git` for *Simple Shapes* and `https://github.com/leopoldmt/Factory.git` for *Factory*.

B Model Parameters

In this Appendix, we provide details about our models' implementation, starting with the β-VAE used in both visual environments: *Simple Shapes* (Table 1) and *Factory* (Table 2). In the VAE encoder, all convolutions have a padding of 1, a stride of 2, and a kernel-size of 4. For the decoders, in Simple Shapes (Table 1), the transposed convolutions have a padding of 1, a stride of 2, and a kernel size of 4, except the first one which has a stride of 1. The final convolution has a stride of 1 and a kernel size of 4. In Factory (Table 2), the transposed convolutions have a padding of 2, a stride of 2, and a kernel size of 5, except the first one which has a stride of 1 and a kernel size of 8 without padding. The final convolution has a stride of 1 and a kernel size of 5. For both environments the β value was set to 0.1. The β-VAE was always trained with the entire set $\mathcal{U}_v$ in both environments (500 000 images in *Simple Shapes* and 200 000 images in *Factory*).

VAE encoder ($2.8M$ params)	VAE decoder ($3M$ params)
$x \in \mathbb{R}^{3\times32\times32}$	$z \in \mathbb{R}^{12}$
Conv_{128} − BN − ReLU	$\text{FC}_{8\times8\times1024}$
Conv_{256} − BN − ReLU	ConvT_{512} − BN − ReLU
Conv_{512} − BN − ReLU	ConvT_{256} − BN − ReLU
Conv_{1024} − BN − ReLU	ConvT_{128} − BN − ReLU
Flatten − $\text{FC}_{2\times12}$	Conv_1 − Sigmoid

Table 1: Architecture and number of parameters of the VAE used in the *Simple Shapes* environment.

VAE encoder ($2.8M$ params)	VAE decoder ($5M$ params)
$x \in \mathbb{R}^{3\times128\times128}$	$z \in \mathbb{R}^{10}$
Conv_{128} − BN − ReLU	$\text{FC}_{8\times8\times512}$
Conv_{256} − BN − ReLU	ConvT_{256} − BN − ReLU
Conv_{512} − BN − ReLU	ConvT_{128} − BN − ReLU
Conv_{1024} − BN − ReLU	ConvT_{64} − BN − ReLU
Flatten − FC_{10}	Conv_1 − Sigmoid

Table 2: Architecture and number of parameters of the VAE used in the *Factory* environment.

Table 3 and Table 4 present details about the Global Workspace architecture for respectively *Simple Shapes* and *Factory*. The tables show the architecture for the encoder and decoder of only one modality, since they are nearly identical across modalities. Only the last Fully Connected layer of the decoders is different, outputting a vector of the original size of each domain.

GW encoder ($35K$ params)	GW decoder ($50K$ params)
FC_{128} − ReLU	FC_{128} − ReLU
FC_{128} − ReLU	FC_{128} − ReLU
FC_{128} − ReLU	FC_{128} − ReLU
FC	FC

Table 3: Architecture and number of parameters for the encoder and decoder in the GW of one modality in *Simple Shapes*

The implementation details for AVAE are presented in Table 5 for *Simple Shapes* and Table 6 for *Factory*. In both environments the parameters for the Conv and ConvT layers in the image VAE are the same as the ones used in their respective VAE in Tables 1 and 2. For *Simple Shapes*, the input

GW encoder ($1.3M$ params)	GW decoder ($1.3M$ params)
$\text{FC}_{512} - \text{ReLU}$	$\text{FC}_{512} - \text{ReLU}$
$\text{FC}_{512} - \text{ReLU}$	$\text{FC}_{512} - \text{ReLU}$
$\text{FC}_{512} - \text{ReLU}$	$\text{FC}_{512} - \text{ReLU}$
$\text{FC}_{512} - \text{ReLU}$	$\text{FC}_{512} - \text{ReLU}$
$\text{FC}_{512} - \text{ReLU}$	$\text{FC}_{512} - \text{ReLU}$
FC	FC

Table 4: Architecture and number of parameters for the encoder and decoder in the GW of one modality in *Factory*

layer of the attributes side is divided in two Fully Connected layers: one for the category of the shape (one-hot vector) and one for the rest of the attributes (continuous values).

AVAE vision ($6M$ params)	AVAE attributes ($0.6M$ params)
$x \in \mathbb{R}^{3\times32\times32}$	$x \in \{0,1\}^3 \times \mathbb{R}^8$
$\text{Conv}_{128} - \text{BN} - \text{ReLU}$	$\text{FC}_{128} - \text{ReLU}$
$\text{Conv}_{256} - \text{BN} - \text{ReLU}$	$\text{FC}_{128} - \text{ReLU}$
$\text{Conv}_{512} - \text{BN} - \text{ReLU}$	$\text{FC}_{12} - \text{ReLU}$
$\text{Conv}_{1024} - \text{BN} - \text{ReLU}$	
$\text{Flatten} - \text{FC}_{2\times12}$	$\text{FC}_{2\times12}$
$z \in \mathbb{R}^{12}$	$z \in \mathbb{R}^{12}$
$\text{FC}_{8\times8\times1024}$	
$\text{ConvT}_{512} - \text{BN} - \text{ReLU}$	$\text{FC}_{128} - \text{ReLU}$
$\text{ConvT}_{256} - \text{BN} - \text{ReLU}$	$\text{FC}_{128} - \text{ReLU}$
$\text{ConvT}_{128} - \text{BN} - \text{ReLU}$	$[\text{FC}_3, \text{FC}_8 - \text{Tanh}]$
$\text{Conv}_1 - \text{Sigmoid}$	

Table 5: Architecture and number of parameters of the visual and attributes VAEs of the AVAE for the *Simple Shapes* environment.

AVAE vision ($11M$ params)	AVAE attributes ($2M$ params)
$x \in \mathbb{R}^{3\times128\times128}$	$x \in \mathbb{R}^{10}$
$\text{Conv}_{128} - \text{BN} - \text{ReLU}$	$\text{FC}_{512} - \text{ReLU}$
$\text{Conv}_{256} - \text{BN} - \text{ReLU}$	$\text{FC}_{512} - \text{ReLU}$
$\text{Conv}_{512} - \text{BN} - \text{ReLU}$	$\text{FC}_{40} - \text{ReLU}$
$\text{Conv}_{1024} - \text{BN} - \text{ReLU}$	
$\text{Flatten} - \text{FC}_{2\times40}$	$\text{FC}_{2\times40}$
$z \in \mathbb{R}^{40}$	$z \in \mathbb{R}^{40}$
$\text{FC}_{8\times8\times1024}$	
$\text{ConvT}_{512} - \text{BN} - \text{ReLU}$	$\text{FC}_{512} - \text{ReLU}$
$\text{ConvT}_{256} - \text{BN} - \text{ReLU}$	$\text{FC}_{512} - \text{ReLU}$
$\text{ConvT}_{128} - \text{BN} - \text{ReLU}$	$\text{FC}_{10} - \text{Tanh}$
$\text{Conv}_1 - \text{Sigmoid}$	

Table 6: Architecture and number of parameters of the visual and attributes VAEs of the AVAE for the *Factory* environment.

C VAE exploration

Figures 4 and 5 illustrate the generation capabilities of each VAE in *Factory* and *Simple Shapes*. To produce these Figures an image was encoded to obtain a latent vector. Each dimension of this vector was modified by adding the value on top of the column keeping the rest frozen. The modified vector was then decoded to obtain a resulting image. The image in the middle column in both Figure represent the initial image encoded in the VAE because the change applied to the vector was null. This technique allows to visualize the information contains in the different dimension. In *Factory*'s

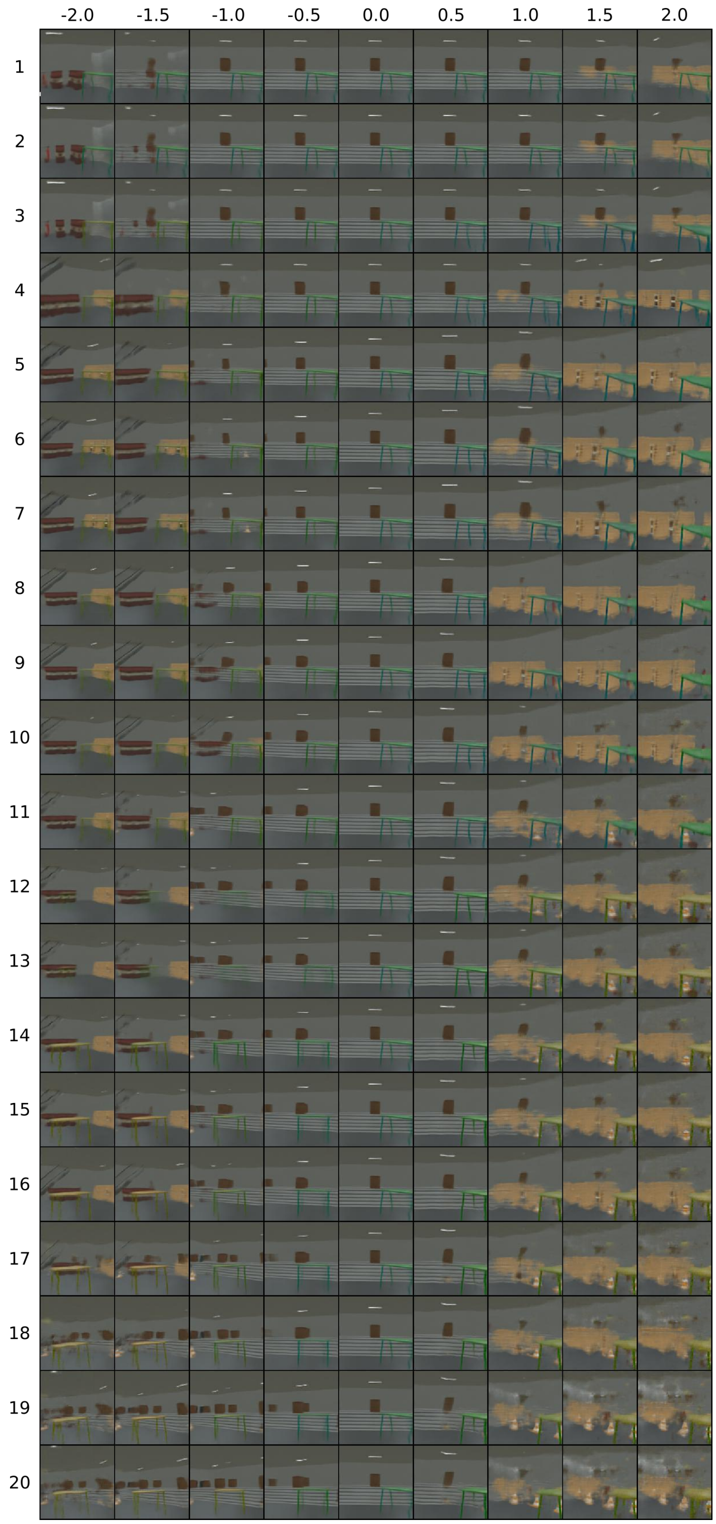

Figure 4: Latent traversal of the VAE used in *Factory*. The rows represent the modified dimension and the columns the value added to the initial before decoding the latent vector.

VAE the background is always recognisable but the table can be blurry. We can still guess its colour and approximate position with respect to the robot viewpoint (moreless on the right, left or in front

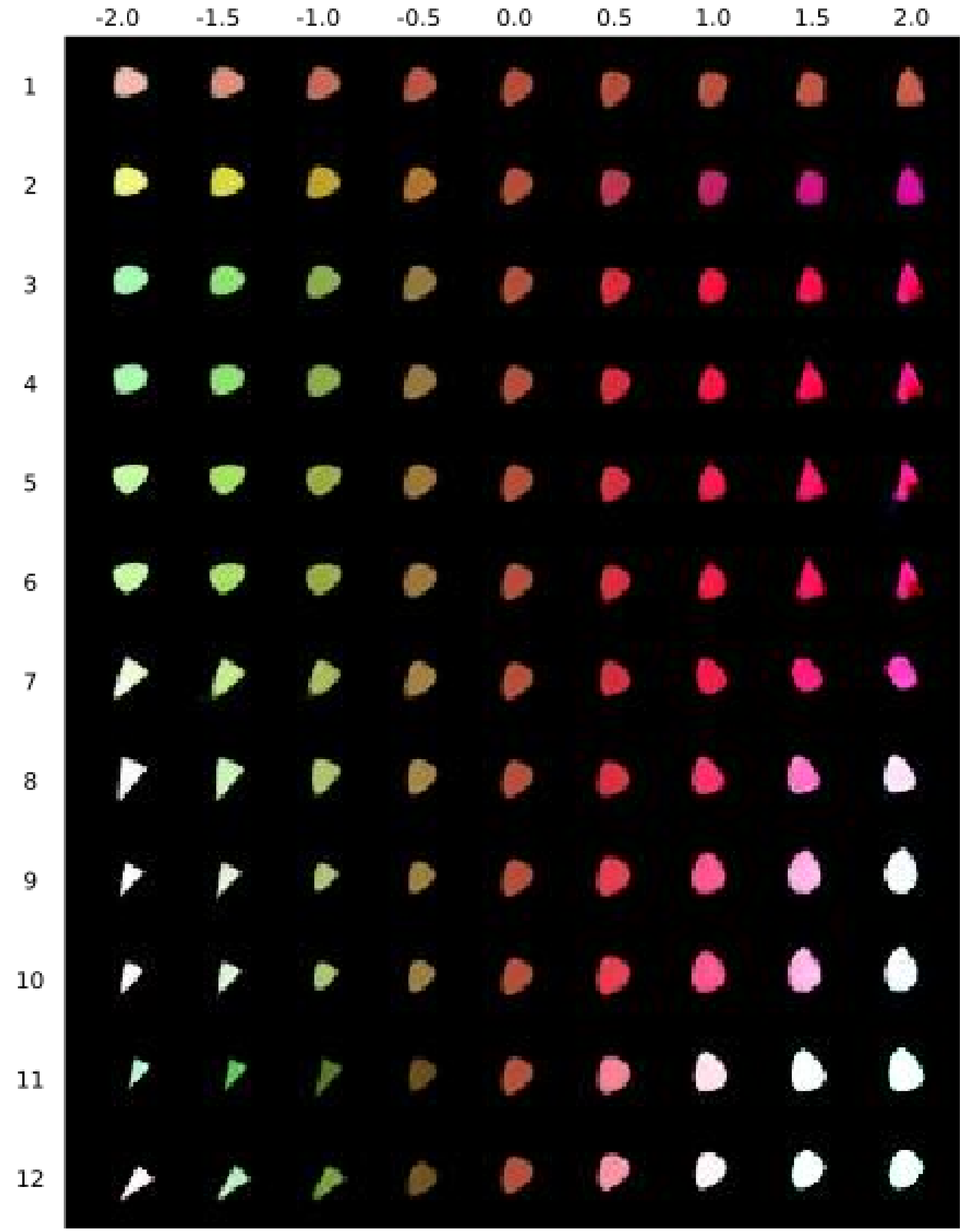

Figure 5: Latent traversal of the VAE used in *Simple Shapes*. The rows represent the modified dimension and the columns the values added to the initial before decoding the latent vector.

of the robot). For *Simple Shapes* the VAE is capable of generating a wide variety of images for this environment covering all possible variations (type of shape, position, rotation, size, color).

D GW losses details

As explained in 3, the Global Workspace (GW) is trained with four different losses. Here we provide details of their implementation, following Devillers et al. [2023].

$$
\begin{aligned}
L_{tr} &= \frac{1}{2}[L_{attr}(d_{attr}(e_v(o_v^i)), o_{attr}^j) + L_v(d_v(e_{attr}(o_{attr}^j)), o_v^i)] \\
L_{cont} &= CONT[e_v(o_v^i), e_{attr}(o_{attr}^j)] \\
L_{dcy} &= \frac{1}{2}[L_v(d_v(e_v(o_v^i)), o_v^i) + L_{attr}(d_{attr}(e_{attr}(o_{attr}^j)), o_{attr}^j)] \\
L_{cy} &= \frac{1}{2}[L_v(d_v(e_{attr}(d_{attr}(e_v(o_v^i)))), o_v^i) + L_{attr}(d_{attr}(e_v(d_v(e_{attr}(o_{attr}^j)))), o_{attr}^j)]
\end{aligned}
$$

Where $CONT()$ is the contrastive loss used in the CLIP model [Radford et al., 2021]. L_{attr} represents the reconstruction loss used on the attributes side, which differs between the two environments. In *Factory* (where all attributes have continuous values), it is computed with an MSE; in *Simple Shapes* it is a combination of a negative log-likelihood for shape classes (discrete one-hot encoded values) and MSE for the other (continuous) attributes. L_v represents the reconstruction loss on the visual side, computed with an MSE in both environments. The total loss is then computed as follows :

$$L_{GW} = \alpha \cdot L_{tr} + \beta \cdot L_{cont} + \gamma \cdot L_{dcy} + \theta \cdot L_{cy}$$

Where $\alpha, \beta, \gamma, \theta$ are hyperparameters giving more or less importance to each loss. The following table contains the hyperparameters for all Global Workspace models (and ablations) in the Full data regime in both environments.

	GW	GW w/o cycles	CLIP-like
Factory	$\alpha = 1$	$\alpha = 1$	$\alpha = 0$
	$\beta = 0.1$	$\beta = 0.1$	$\beta = 1$
	$\gamma = 1$	$\gamma = 0$	$\gamma = 0$
	$\theta = 1$	$\theta = 0$	$\theta = 0$
Simple Shapes	$\alpha = 1$	$\alpha = 1$	$\alpha = 0$
	$\beta = 0.1$	$\beta = 0.1$	$\beta = 1$
	$\gamma = 5$	$\gamma = 0$	$\gamma = 0$
	$\theta = 5$	$\theta = 0$	$\theta = 0$

The table below shows the hyperparameters used in the Low data regime in both environments.

	GW	GW w/o cycles	CLIP-like
Factory	$\alpha = 1$	$\alpha = 1$	$\alpha = 0$
	$\beta = 0.1$	$\beta = 0.1$	$\beta = 1$
	$\gamma = 5$	$\gamma = 0$	$\gamma = 0$
	$\theta = 5$	$\theta = 0$	$\theta = 0$
Simple Shapes	$\alpha = 1$	$\alpha = 1$	$\alpha = 0$
	$\beta = 0.1$	$\beta = 0.1$	$\beta = 1$
	$\gamma = 10$	$\gamma = 0$	$\gamma = 0$
	$\theta = 10$	$\theta = 0$	$\theta = 0$

E Reward details

The reward in the *Factory* environment is given by a combination of the distance between the robot and the table, and the angle between the orientation of the robot and the table (this is meant to encourage the policy to turn the robot facing the table, regardless of its original location):

$$r = -\text{distance} - 10 \times \text{angle}$$

$$r = -\sqrt{(x_r - x_t)^2 + (y_r - y_t)^2} - 10 \times |\arccos([c_{\theta_r}, s_{\theta_r}], \frac{[x_t - x_r, y_t - y_r]}{||[x_t - x_r, y_t - y_r]||_2})|$$

The reward in the *Simple Shapes* environment is given by a combination of the distance between the position of the shape and the center of the image, and the angle of the shape:

$$r = -\text{distance} - 10 \times \text{angle}$$

$$r = -\sqrt{(x - 16)^2 + (y - 16)^2} - 10 \times |\arccos([c_\theta, s_\theta], [1, 0])|$$

F Computational Resources

All the models training were performed on local NVIDIA Quadro RTX 5000 with 16Gb of RAM. The two VAEs (one for *Factory* and one for *Simple Shapes*) and the AVAE training took 2 days each. The training duration of the Global Workspace depends on the losses hyperparameters but took in average 2 hours. Finally the agent for both environment was trained with a maximum of 1 million steps. In *Factory* the agent was trained during 6 hours on NVIDIA Quadro RTX 5000. For *Simple Shapes* the training was performed on NVIDIA A100 80GB during 8 hours.

G A2C in Simple Shapes scenario

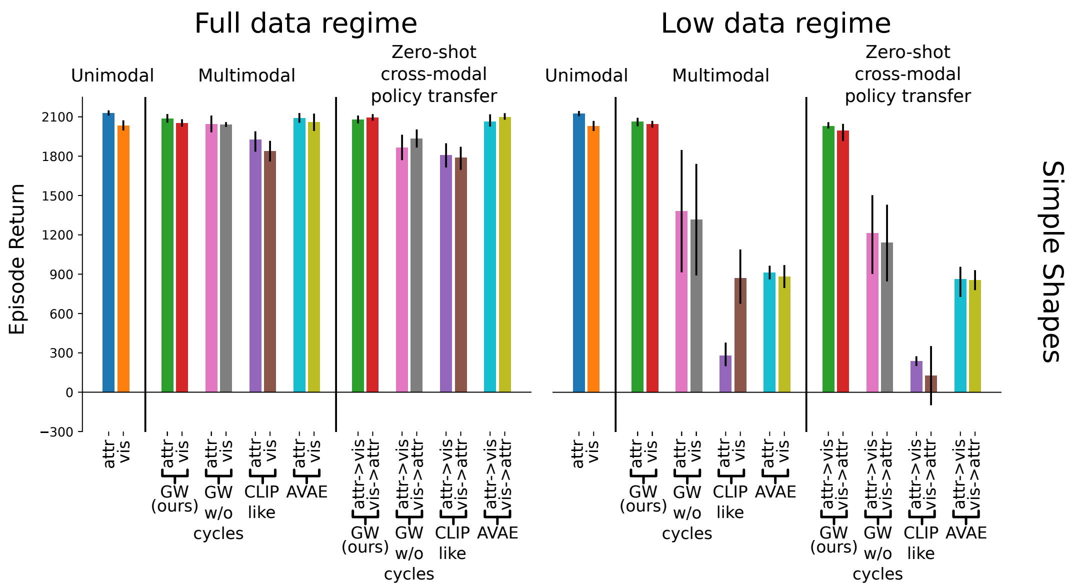

Figure 6: Performance of A2C in the Simple Shapes environment. Notations and conventions as in Figure 3.

An additional experiment was performed in the Simple Shapes environment to verify that our results were robust to the choice of policy training algorithm. For this, we used A2C, introduced by Mnih et al. [2016]. Figure 6 shows that the results are reproducible with this alternative algorithm (compare with Figure 3, bottom). A2C trained from a Global Workspace performs as well as when trained on unimodal representations, both in terms of absolute performance and in terms of zero-shot cross-modal transfer. AVAE performs similarly in the Full data regime, but poorly in the Low data regime. The two other models (Global Workspace without cycles and CLIP-like ablation), give worse performance in both regimes, as in the case of PPO.