# Multimodal Dreaming: A Global Workspace Approach to World Model-Based Reinforcement Learning

# Léopold Maytié (leopold.maytie@univ-tlse3.fr)

CerCo, CNRS UMR5549 Artificial and Natural Intelligence Toulouse Institute Université de Toulouse

#### Roland Bertin Johannet (roland.bertin-johannet@cnrs.fr)

CerCo, CNRS UMR5549 Université de Toulouse

## Rufin VanRullen (rufin.vanrullen@cnrs.fr)

CerCo, CNRS UMR5549

Artificial and Natural Intelligence Toulouse Institute
Université de Toulouse

# **Abstract**

Humans leverage rich internal models of the world to reason about the future, imagine counterfactuals, and adapt flexibly to new situations. In Reinforcement Learning (RL), world models aim to capture how the environment evolves in response to the agent's actions, facilitating planning and generalization. However, typical world models directly operate on the environment variables (e.g. pixels, physical attributes), which can make their training slow and cumbersome; instead, it may be advantageous to rely on high-level latent dimensions that capture relevant multimodal variables. Global Workspace (GW) Theory offers a cognitive framework for multimodal integration and information broadcasting in the brain, and recent studies have begun to introduce efficient deep learning implementations of GW. Here, we evaluate the capabilities of an RL system combining GW with a world model. We compare our GW-Dreamer with various versions of the standard PPO and the original Dreamer algorithms. We show that performing the dreaming process (i.e., mental simulation) inside the GW latent space allows for training with fewer environment steps. As an additional emergent property, the resulting model (but not its comparison baselines) displays strong robustness to the absence of one of its observation modalities (images or simulation attributes). We conclude that the combination of GW with World Models holds great potential for improving decision-making in RL agents.

**Keywords:** World Models; Global Workspace Theory; Reinforcement Learning; Multimodal Representation Learning; Mental Simulation

#### Introduction

Humans possess the ability to anticipate the consequences of their actions before executing them in the real world. This capacity suggests that humans construct an internal World Model (see e.g. Friston (2010); Clark (2013), among others).

In artificial intelligence (AI), this concept has been particularly applied in World Model-based reinforcement learning (RL), a subset of model-based RL. In model-based RL, transition dynamics of the environment are traditionally specified as Markov decision processes (MDPs), either manually defined (Sutton, 1991; Atkeson & Santamaria, 1997) or empirically estimated from interaction data (Dearden et al., n.d.; Szita & Szepesvári, n.d.). While model-based RL is generally more sample-efficient than model-free RL, constructing accurate transition models remains challenging. Learning a World Model directly from data facilitates decision-making in environments where transition dynamics are either unknown or too complex to specify explicitly. This idea was popularized by Ha & Schmidhuber (2018) and later extended by the Dreamer framework (Hafner et al., 2024), allowing agents to learn by performing mental simulations of episodes rather than relying solely on direct interaction with the environment.

More generally, AI research on world models has gained further momentum with the advent of large-scale World Foundation Models, such as Genie (Bruce et al., 2024) trained in an unsupervised manner, or World Foundation Models platforms like Cosmos from Nvidia (NVIDIA et al., 2025)

Beyond their ability to anticipate the consequences of their actions, humans perceive the world through multiple sensory modalities, leading to a rich and robust representation of their environment. The Global Workspace Theory (GWT), introduced by Baars (1988) and later expanded by Dehaene et al. (1998), provides a framework to explain such integrative cognitive processes. According to this theory, specialized modules compete to encode their information into a shared space called the Global Workspace. Through a broadcasting mechanism, this information becomes accessible to various brain regions, shaping our conscious experience.

Theoretical proposals have linked World Models and Global Workspace Theory (GWT). VanRullen & Kanai (2021) align closely with GWT, suggesting a World Model module that interacts with a shared representational space. Similarly, the Integrated World Model Theory (IWMT) Safron (2022) seeks

to unify these concepts by incorporating predictive and generative mechanisms for structuring internal representations. While both emphasize information integration, GWT focuses on selective broadcasting for decision-making and awareness, whereas IWMT prioritizes constructing an internal model for prediction and planning. Inspired by these frameworks, our work explores multimodal integration and predictive mechanisms without committing to (or rejecting) their assumptions about consciousness.

In this paper we introduce a system bridging the ideas of World Model and Global Workspace (VanRullen & Kanai, 2021; Safron, 2022). We took inspiration from the architecture proposed in Dreamer algorithms (Ha & Schmidhuber, 2018; Hafner et al., 2024) and extended the Global Workspace implementation proposed by Devillers et al. (2024) to implement our Global Workspace Dreamer (GW-Dreamer). The key originality of GW-Dreamer is that it learns to represent the World-Model transitions using multimodal GW representations. We compared our model in two different Reinforcement Learning environments against standard PPO and the original Dreamer algorithm as well as a variant of Dreamer that shared the same visual input module as GW-Dreamer. Thanks to its efficient GW multimodal latent representation, our model learns with fewer environment interactions; in addition, it proves more robust to missing modalities (as already shown in the context of model-free RL by Maytié et al. (2024)).

# Model

In this study, we consider RL environments with multimodal observations. By consequence, the state of an environment at time t leads to multiple observations  $o_t \in \mathcal{O}$ , which can be either an RGB image  $o_t^v$ , or an attribute vector  $o_t^{attr}$  describing physical attributes of the simulation. From these observations, the agent predicts an action  $a_t \in \mathcal{A}$  to interact with the environment, leading to a reward  $r_{t+1}$ .

To interact with these multimodal environments we propose a model composed of three main components: a representation model, called Global Workspace, a World Model and an Actor-Critic RL policy. The training process consists of two main steps. First, the Global Workspace is trained to represent the multimodal environment using a dataset of environment observations collected randomly or via an expert agent (Figures 1,2). Then, the World Model and the Actor-Critic are trained through interaction with the environment (Figure 3). In the following subparts, we provide a detailed description of the architecture and training procedure for each component.

# **Global Workspace**

The Global Workspace serves as a representation model, encoding multiple modalities into a unified latent representation. This latent representation is then used by the World Model to encapsulate the agent's perception of the environment at time t. Our proposed Global Workspace is inspired by the approach introduced by Devillers et al. (2024). However, we modify both the architecture and training procedure to produce a single unified representation, rather than maintaining

separate representations for each modality. This architecture and its training losses are illustrated in Figure 1.

As proposed by VanRullen & Kanai (2021) and implemented by Devillers et al. (2024), we do not train our set of encoders and decoders directly from raw modalities. Instead, we employ two pretrained and frozen modules (in this case, VAEs) to transform raw representations (denoted  $o^{\nu}$  for images and  $o^{attr}$  for attributes) into unimodal latent representations ( $u^{\nu}$  and  $u^{attr}$ ). These unimodal representations are then encoded into pre-fusion latent variables  $z^{\nu} = e_{\nu}(u^{\nu})$  and  $z^{attr} = e_{attr}(u^{attr})$ . However, in contrast to the model proposed by Devillers, we do not always directly decode from these prefusion representations. We combine them using element-wise weighted sum followed by a Tanh activation function to form a unified representation denoted z. From this unified representation, we can recover the unimodal representations through a set of decoders  $d_{\nu}$  and  $d_{attr}$ .

The GW model is trained using two loss functions: the contrastive loss and the broadcast loss. The contrastive loss  $\mathcal{L}_{cont}$  follows a similar formulation to the one proposed in CLIP (Radford et al., 2021); it is designed to align the representations  $z^{v}$  and  $z^{attr}$  before fusion, supporting the development of amodal representations in the Global Workspace. The broadcast loss is inspired by the broadcast principle at the heart of GWT; it is computed by comparing the predicted  $(\hat{u}^v, \hat{u}^{attr})$ and ground-truth unimodal representations ( $u^{v}$ ,  $u^{attr}$ ) using the mean squared error (MSE). Specifically, it consists of a weighted sum of multiple sub-losses, including the cycle loss  $(\mathcal{L}_{cv})$ , demi-cycle loss  $(\mathcal{L}_{dcv})$ , and translation loss  $(\mathcal{L}_{tr})$ , as introduced by Devillers et al. (2024). Additionally, a fusion loss ( $\mathcal{L}_{fusion}$ ) is incorporated. The primary objective of these losses is to ensure that the unimodal latent vectors can be accurately reconstructed after the weighted-sum fusion of pre-Global Workspace representations ( $z^{v}$  and  $z^{attr}$ ), regardless of the exact fusion weights employed. The weighting factors  $\alpha_{attr}$  and  $\alpha_{\nu}$  are adjusted for each specific sub-loss (and with the constraints  $\alpha_{\nu} \geq 0, \alpha_{attr} \geq 0, \alpha_{\nu} + \alpha_{attr} = 1$ ), as defined below. (For an in-depth discussion of the usefulness of each loss term  $\mathcal{L}_{cont}$ ,  $\mathcal{L}_{cy}$ ,  $\mathcal{L}_{dcy}$  and  $\mathcal{L}_{tr}$ , see Devillers et al. (2024)).

$$\forall (x,y) \in \{attr,v\}, \quad x \neq y$$

$$\begin{cases}
\mathcal{L}_{dcy} = \|d_x(tanh(e_x(u^x))) - u^x\|_2^2, \\
(\alpha_x = 1, \alpha_y = 0)
\end{cases}$$

$$\mathcal{L}_{cy} = \|d_x(tanh(e_y(d_y(tanh(e_x(u^x)))))) - u^x\|_2^2, \\
(\alpha_x = 1, \alpha_y = 0), \text{ then } (\alpha_x = 0, \alpha_y = 1)
\end{cases}$$

$$\mathcal{L}_{fr} = \|d_y(tanh(e_x(u^x))) - u^y\|_2^2, \\
(\alpha_x = 1, \alpha_y = 0)
\end{cases}$$

$$\mathcal{L}_{fusion} = \|d_x(tanh(\alpha_x.e_x(u^x) + \alpha_y.e_y(u^y))) - u^x\|_2^2, \\
(\alpha_x > 0, \alpha_y > 0, \alpha_x + \alpha_y = 1)
\end{cases}$$
(1)

$$\mathcal{L}_{broad} = \beta_{dcv}.\mathcal{L}_{dcv} + \beta_{cv}.\mathcal{L}_{cv} + \beta_{tr}.\mathcal{L}_{tr} + \beta_{fusion}.\mathcal{L}_{fusion}$$
 (2)

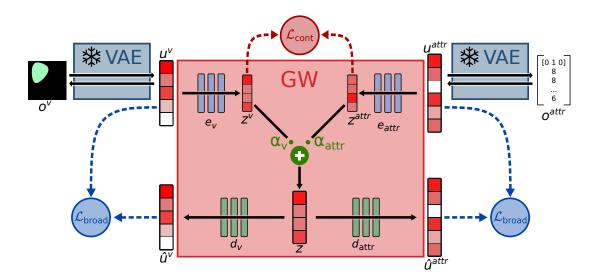


Figure 1: Overview of the Global Workspace model for multimodal representation. Raw environment inputs (image pixels, simulation attributes) are encoded in their latent unimodal representation ( $u^v$  or  $u^{attr}$ ) thanks to pretrained (and frozen) VAEs. These unimodal latent representations are then processed by encoders  $e_v$  and  $e_{attr}$  (respectively) to produce pre-GW representations ( $z^v$  and  $z^{attr}$ ). The final Global Workspace representation  $z \in Z$  is obtained by fusing these pre-GW representations through an element-wise weighted sum (with weights  $\alpha_v \geq 0$  and  $\alpha_{attr} \geq 0$ ,  $\alpha_v + \alpha_{attr} = 1$ ) followed by a Tanh activation. The unimodal latent vectors can be retrieved from z with a set of decoders  $d_v$  and  $d_{attr}$ . The GW component networks  $e_v$ ,  $e_{attr}$ ,  $d_v$  and  $d_{attr}$  are trained by combining a contrastive loss  $\mathcal{L}_{cont}$  and a broadcast  $\mathcal{L}_{broad}$ . The former encourages the pre-GW representations to align across modalities; the latter also promotes this objective (see Devillers et al. (2024)), and ensures that decoded or "broadcasted" GW representations resemble the original unimodal latent representations, regardless of each modality's initial contribution to the GW representation (as captured by the fusion weights  $\alpha_v$  and  $\alpha_{attr}$ ). Once trained, the GW model is frozen for learning the World Model and RL policy (Figure 3), and the fusion weights are fixed ( $\alpha_v = \alpha_{attr} = 0.5$ ).

The full training objective of the Global Workspace is a weighted sum of the contrastive loss and the broadcast loss, as shown below. The weight of the broadcast loss is fixed at one (its overall contribution can be adjusted by modifying the individual weights that constitute it:  $\beta_{dcy}$ ,  $\beta_{cy}$ ,  $\beta_{tr}$ ,  $\beta_{fusion}$ ).

$$\mathcal{L}_{GW} = \beta_{cont} \cdot \mathcal{L}_{cont} + \mathcal{L}_{broad}$$
 (3)

Figure 2 illustrates the functional properties of the trained GW when processing multimodal inputs. Here, for illustration purposes, two images (left) are chosen to differ from the attributes vector (right) along the color and size dimensions. By modulating the fusion coefficients  $\alpha_v$  and  $\alpha_{attr}$ , the GW can perform distinct functional operations. For instance, it can perform attribute-to-image translation by setting the fusion coefficient to  $\alpha_v = 0$  and  $\alpha_{attr} = 1$ . This configuration ensures that only the attributes representation propagates through the GW, while visual information is disregarded. The following decoding step (using  $d_v$  and the visual VAE), reconstructs an image that is inferred purely from the attribute representation. The results, shown above the "tr" label in Figure 2, confirm that the reconstructed image remains unaffected by the suppressed visual input. A second operational mode, referred to as a demi-cycle, is illustrated in the "dcy" section of Figure 2. Here, the GW selectively propagates only the visual representation by setting  $\alpha_v = 1$  and  $\alpha_{attr} = 0$ . The reconstructed images exclusively reflect the visual inputs. Beyond unimodal processing, the GW also supports multimodal fusion, enabling the integration of heterogeneous information streams. In the "fusion" section of Figure 2, both modalities are encoded into the GW with equal weighting ( $\alpha_v = 0.5, \alpha_{attr} = 0.5$ ), allowing information from both sources to be jointly encoded in the shared latent space. Decoding this fused representation results in an image that integrates features from both the original visual input and attributes: the reconstructed color and size are halfway between those of the attributes and image inputs. Thus, by appropriately tuning the fusion coefficients following encoding  $(e_v, e_{attr})$  and by selecting the appropriate decoding pathway ( $d_v$  or  $d_{attr}$ ), one can dynamically reconfigure the functional role of the GW. Specifically, it can transition between unimodal reconstruction, cross-modal translation, and multimodal fusion, providing a flexible framework for integrating and transforming heterogeneous information sources.

#### **World Model**

The World Model (WM) is a fundamental component that enables to train the RL agent through a mental simulation or "dreaming" process. At each time step t, this recurrent network (with internal hidden state  $h_t$ ) receives as input the GW representation ( $z_t$ ) and an action  $a_t$ , from which it predicts the

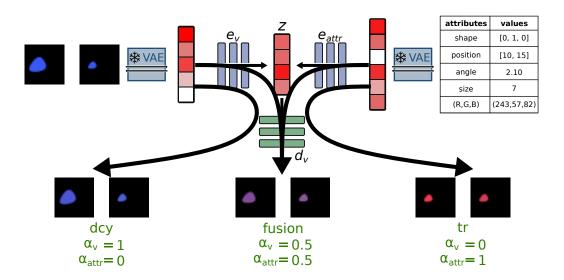


Figure 2: Illustration of the behaviour of the GW. We start from a fixed attribute vector describing a small red egg-shape (top right), and two images (top left) that are chosen to differ in terms of color (right-most image) or both color and size (left-most image). These inputs are encoded into the GW using different fusion weights  $\alpha_v$  and  $\alpha_{attr}$ , indicated in green below each configuration, and subsequently decoded into an image. The resulting images at the bottom illustrate three distinct functional modes of operation. In the translation mode (tr, bottom right), both modalities are encoded, but only attribute information is transmitted through the GW, while visual input is disregarded. The reconstructed images, obtained by decoding the GW latent vector z as an image using  $d_v$  and the visual VAE, demonstrate the successful translation of attribute information into the visual domain. In the demi-cycle mode (dcy, bottom left), both modalities are encoded, but only the visual information is propagated through the GW. The absence of distortions due to attributes information in the reconstructed images confirms that attribute information was effectively suppressed. In the fusion mode (bottom middle), both modalities are encoded with equal weights, allowing information from both sources to be integrated inside the GW. The decoded images reflect a hybrid representation of vision and attributes features, resulting in an intermediate color and size.

GW representation at the next time step  $(z_{t+1})$ , the reward associated with the action  $(r_{t+1})$ , and a Boolean termination signal  $(d_{t+1})$  indicating whether the task is complete. The GW representation is computed from environmental observations using the pre-trained (and frozen) GW model described above. The model employs a Gated Recurrent Unit (GRU) (Cho et al., 2014) for recurrence, while the prediction heads consist of a set of Multi-Layer Perceptrons (MLPs).

The World Model is trained on data collected from the environment using the current policy of the Actor (see below), while keeping the GW model frozen. Its objective is to predict the GW representation, reward, and termination flag at the next time step (t+1). The corresponding loss function ( $\mathcal{L}_{WM}$ ) is computed as a weighted sum, combining MSE for the predicted GW representation and reward, with Binary Cross-Entropy (BCE) for the termination flag. This is illustrated in Figure 3 (1).

## **Actor-Critic**

The Actor policy is learned concurrently with the World Model in alternating steps, as detailed below. Initially, n data pairs  $(o_t, a_t, o_{t+1}, r_{t+1}, d_{t+1})$  are collected through interaction with the environment and stored in a replay buffer. Once the data is gathered, m learning steps are performed. The learn-

ing alternates between training the World Model as described previously and the Actor-Critic network. The Actor-Critic takes as input the internal representation from the GRU,  $h_t$ , and predicts both the action  $a_t$  and the state value  $v_t$ . This approach closely follows the methodology proposed in Dreamer (Hafner et al., 2024). It learns from "mental simulations" or "dreaming" instead of interacting directly with the environment. During training, the observations are provided only at the first step. For the following ones, the World Model simulates the environment for a predetermined number of steps, using the actions predicted by the Actor, without access to true observations, as shown in Figure 3 (2). During AC training, the gradient does not propagate through the World Model, ensuring that the learned policy does not directly influence the internal dynamics of the World Model (and vice-versa, WM training gradients do not propagate to the AC network). The simulated actions and values, along with the predicted rewards and done signals, are used in the loss function of the Actor-Critic network (following the standard Actor-Critic algorithm (Konda & Tsitsiklis, 1999) modified in Dreamer (Hafner et al., 2024)). This entire training procedure is described explicitly in Algorithm 1.

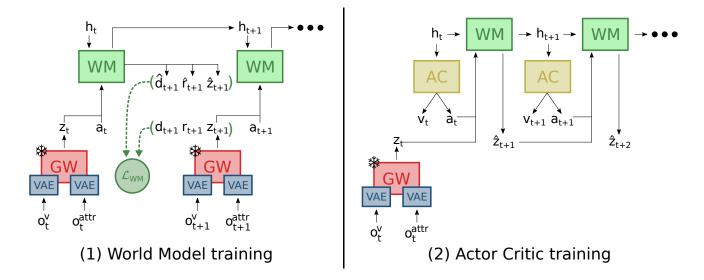


Figure 3: (1) World Model training: At each time step, the environment provides observations  $(o_t^v, o_t^{attr})$ , a reward  $r_t$ , and a termination signal  $d_t$ . A pretrained and frozen Global Workspace (GW) model, incorporating a Variational Autoencoder (VAE) for each modality, encodes observations into a GW representation  $z_t$ . The WM is trained on sequences of data collected from the environment using the current AC policy. Given  $z_t$  and the action  $a_t$  predicted by the policy, the WM (implemented as a GRU: Gated Recurrent Unit) updates its internal state from  $h_t$  to  $h_{t+1}$ . Using this updated state, the WM predicts the next GW representation  $z_{t+1}$ , the expected reward  $r_{t+1}$ , and the termination signal  $d_{t+1}$  with three separate prediction heads. The loss function  $\mathcal{L}_{WM}$  is computed as a weighted sum of the Mean Squared Error (MSE) for  $z_{t+1}$  and  $r_{t+1}$ , and the Binary Cross-Entropy (BCE) loss for predicting  $d_{t+1}$ . (2) Actor-Critic training: The AC model is trained using "mental simulation". The GW representation  $z_t$  derived from observations is provided only at the first time step. For subsequent steps, the WM generates novel states by processing the previously predicted GW representation and the action selected by the AC. The AC loss functions are computed exclusively from the predicted elements within the simulated trajectory, including the generated termination signal  $\hat{d}$ , reward  $\hat{r}$ , and actions taken based on the latent state h.

# Simple Shapes Environment

The different models were tested in an environment called 'Simple Shapes'. This environment was introduced in Devillers et al. (2024) as a fixed dataset, and in Maytié et al. (2024) as an RL environment.

The Simple Shapes environment is multimodal, the agent can receive two types of observations:  $32 \times 32$  pixel RGB images of a 2D shape on a black background, or a set of eight attributes directly describing the environment's state (Figure 4). There are three different types of shapes, an egg-like shape, an isosceles triangle, and a diamond. The shapes possess different properties: 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 agent does not observe these properties directly, but instead receives transformed attributes as observations: the rotation angle  $\theta$  is decomposed into  $(c_\theta, s_\theta) = (cos(\theta), sin(\theta))$ ; HSL colors are translated to the RGB domain, finally, the shape variable is expressed as a one-hot vector of size three, and all variables are normalized between -1 and 1.

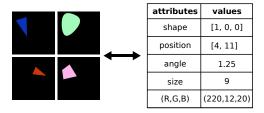
At the beginning of each episode, attributes are randomly sampled within their respective domains; the starting point is thus a random shape of a random orientation, located somewhere in the image. The agent's goal is to move the shape to the center of the image and align it to point to the top. For this purpose, six different actions are available to the agent: 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 the smallest angle (in radians) between the shape's orientation and the null angle times a rotation reward coefficient equal to 10 by default. The episode ends when the shape reaches the goal state, with no additional reward.

#### Results

We evaluated the performance of the GW-Dreamer model against different PPO and Dreamer variants. GW and VAE components were identical across all models and were pretrained using data randomly sampled from the environment. Specifically, we trained: (1) a standard PPO model using only visual inputs  $(o^{\nu})$ , (2) a "VAE-PPO" that uses concatenated representations from both attribute and image VAEs, and (3) a "GW-PPO" that employs a single input representation coming from the GW. This model is expected to have certain advantages relative to the previous two baselines, owing to its strong multimodal abilities (as shown by Maytié et al. (2024)),

# Algorithm 1 GW-Dreamer Training Procedure

- 1: Require Pretrained Global Workspace (GW)
- 2: Initialize World Model (WM), Actor-Critic (AC), Replay Buffer
- 3: while not max number environment steps do
- 4: Collect n transitions  $(o_t, a_t, o_{t+1}, r_{t+1}, d_{t+1})$  from the environment and store in Replay Buffer
- 5: **for** *m* training steps **do**
- 6: Train World Model:
- 7: Transform Replay Buffer observations  $(o_t^v, o_t^{attr})$  into GW latent vectors  $(z_t)$
- 8: Predict next GW representation, reward and done signal:  $WM(z_t, a_t) = (\hat{d}_{t+1}, \hat{r}_{t+1}, \hat{z}_{t+1})$
- 9: Compute  $\mathcal{L}_{WM}$  by comparing against  $(d_{t+1}, r_{t+1}, z_{t+1})$
- 10: Update WM by backpropagating  $\mathcal{L}_{WM}$
- 11: Train Actor-Critic:
- 12: Encode the first observation through GW to get latent representation  $z_t$
- 13: Generate imagined trajectories using WM and the actions predicted by AC
- 14: Update AC using simulated rewards and transitions
- 15: end for
- 16: end while



Actions :

Goal: Place shape at the center pointing to the top

Figure 4: Illustration of the Simple Shapes environment and the task used in this study. The figure presents examples of raw observations, including four example images (left) and one example set of attributes (right). The agent's goal is to place the shape at the center and pointing upward. The agent can move the shape one pixel at a time in four directions (up, down, left, right) or rotate it clockwise or counterclockwise.

but it does not include a World Model. Additionally, we trained (4) the standard Dreamer algorithm using both attribute and image modalities and (5) a "VAE-Dreamer" model that receives VAE-based representations of attributes and images as inputs. The motivation for incorporating latent representations (VAEs) was to enable the models to operate in a fully

latent space, reducing the high compute associated with reconstructing images through decoders. At the same time, this made the underlying architecture closer to GW-Dreamer, except for the absence of a Global Workspace.

All models were trained in the Simple Shapes environment, and the results are illustrated in Figure 5. Returns (cumulative sum of rewards) were normalized such that a return of zero corresponds to the performance of a random agent. Additionally, a "return criterion" was defined as 75% of the highest smoothed reward obtained by any of the models in the figure. We verified visually that this criterion corresponds to a level of return at which the task begins to be solved efficiently.

Figure 5 presents several key findings. First, when comparing different PPO variants, we observe that both multimodal PPO models (VAE-PPO and GW-PPO) reach the return criterion significantly earlier than the standard PPO model. While the standard PPO model meets the criterion just before 1,000,000 steps, VAE-PPO reaches it at 400,000 steps, and GW-PPO at only 200,000 steps. This suggests that incorporating multimodal latent representations can greatly accelerate policy learning, with GW representations yielding even faster convergence than VAE representations.

Second, when comparing Dreamer-based models to PPO-based ones, Dreamer models generally reach the return criterion earlier, corroborating previous findings that world models improve sample efficiency Hafner et al. (2024). In addition, GW-PPO exhibits sample efficiency comparable to both standard Dreamer and VAE-Dreamer, with all three models reaching the threshold at approximately 200,000 environment steps. This suggests that a strong multimodal representation (GW) can bring as much to the model's efficiency as a World Model could. Can the two advantages (GW and World Model) be combined to yield an even more efficient architecture?

This is what Figure 5 seems to suggest: the GW-Dreamer model outperforms all other models, reaching the criterion in just 20,000 steps, i.e. about 10X faster than standard Dreamer and VAE-Dreamer. Since both GW-Dreamer and VAE-Dreamer operate in a multimodal latent space, this result directly highlights the effectiveness of the GW representation in improving sample efficiency within a world-model framework. These findings suggest that the GW representation enables more efficient learning when combined with a dreaming-based approach.

One advantage of training a policy from two input modalities (images and attributes) is that the resulting agent could prove particularly robust in conditions where one of the two modalities becomes noisy or unreliable. We thus conducted an additional experiment to evaluate the zero-shot robustness of GW-Dreamer compared to other multimodal variants when one sensory modality is removed. This scenario simulates real-world conditions where a robot may experience sensor failure or a human may lose one of their senses. Once the models were trained, their parameters were frozen, and we systematically removed either the attribute or image inputs. For models using a GW representation, the fusion mecha-

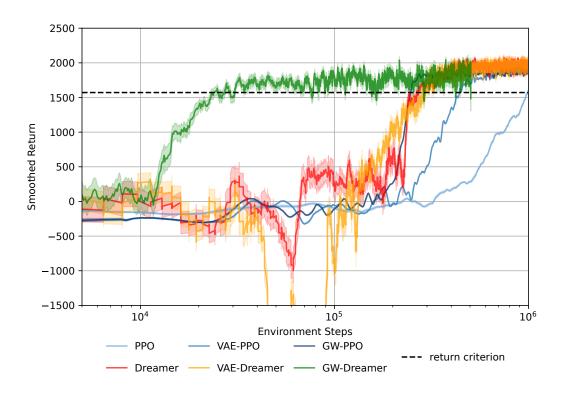


Figure 5: Performance (cumulative sum of rewards or "return") as a function of the number of environment steps (log scale) during training. A fixed baseline, corresponding to the performance of a fully random policy, was subtracted from the episode returns. Thus, a random policy's performance is equal to zero. The returns are smoothed using a sliding window of length 10, with the shaded region indicating the standard error of the mean over this window. The return criterion is defined as 75% of the maximum smoothed return. It corresponds (as verified visually) to a performance at which the task starts to be solved properly.

nism was adjusted accordingly: if attributes were removed, full weight was assigned to the visual modality ( $\alpha_v = 1, \alpha_{attr} = 0$ ), and conversely, if vision was removed the opposite adjustment was made ( $\alpha_v = 0, \alpha_{attr} = 1$ ).

The results, shown in Figure 6, reveal a striking contrast in the robustness of the models. Models using other representations than a GW (VAE-PPO, VAE-Dreamer and Dreamer) showed a complete performance collapse when a modality was removed, indicating a strong dependence on both sensory inputs. In contrast, models using a GW representation (GW-PPO and GW-Dreamer) were the only models to maintain performance above the return criterion and remaining close to their original performances. This demonstrates that GW-Dreamer is capable of solving the task even when one modality is missing, highlighting the importance of a multimodal representation like GW that effectively integrates different sensory inputs. (GW-PPO also benefited from this GW representation, but was less sample-efficient than GW-Dreamer).

These findings underscore the advantages of combining the Global Workspace framework with a World Model. By leveraging this integration, RL algorithms can be trained efficiently while maintaining robustness to modality perturbations, making them more suitable for real-world applications where sensory failures may occur.

# **Discussion and Conclusion**

This paper represents a first step in bridging Global Workspace Theory and World Models in Al. It builds upon the architecture proposed by VanRullen & Kanai (2021) and implemented by Devillers et al. (2024), adapting it to be compatible with World-Model-based reinforcement learning algorithms such as Dreamer (Ha & Schmidhuber, 2018; Hafner et al., 2024).

However, some limitations remain. In terms of absolute performance (measured by return), GW-Dreamer slightly underperforms compared to the original Dreamer algorithm. This difference is somewhat offset, of course, by the improved sample-efficiency and robustness of GW-Dreamer. Additionally, this study relied on a pre-trained GW model trained on randomly sampled data, which may not always be representative of an agent's environment. In more complex environments, expert-collected data would likely be necessary to ensure sufficient coverage of states that are difficult to reach by chance. An alternative approach could involve training the GW representation jointly with the rest of the model. How-

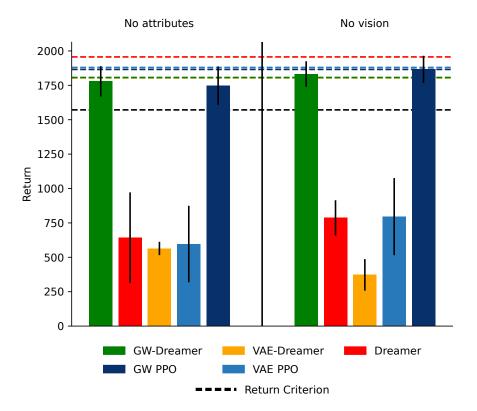


Figure 6: Performance (return) of the different models when one sensory modality is removed. The left part shows results when the attributes modality is discarded, while the right corresponds to the removal of the vision modality. The return criterion is the same as in Figure 5, indicating the performance level at which the task is reliably solved. Dashed colored lines represent the maximum performance of each model from Figure 5, with line and bar colors corresponding to the same models. Performance values are averaged over 10 trials, with error bars representing the standard error of the mean.

ever, the GW itself requires pre-trained latent representations as inputs (VanRullen & Kanai, 2021), modeled here as a VAE; pre-training this latent input space would still potentially lead to the same challenge of appropriate environment sampling, even if the GW itself was trained jointly with the World Model and Actor-Critic components. One way of solving this problem could be to either use a pretrained foundation model capable of encoding arbitrary images in a latent space (Oquab et al., 2023) or to create our own "foundation" encoder trained on large-scale open datasets that have already been collected, e.g. in robotics (Walke et al., 2023; Collaboration et al., 2024)

Another limitation of the present study is that the advantages of GW-Dreamer were demonstrated here in a single, relatively simple test environment. An important direction for future work will be to evaluate the model in more complex scenarios, such as robotic environments.

Despite these limitations, this study demonstrates the advantages of integrating GWT and WM. This combination significantly improves training efficiency in RL, allowing GW-Dreamer to learn with about 10 times fewer environment steps. This accelerated training is not due to operating entirely in a latent space, as GW-Dreamer also surpasses (in terms of

sample efficiency) a Dreamer variant that relies on latent VAE representations. Furthermore, GW-Dreamer displays zero-shot robustness to modality loss. Unlike the original Dreamer, GW-Dreamer maintains its performance even when visual inputs or attribute information are removed (similarly to findings obtained for model-free RL algorithms in Maytié et al. (2024), and corroborated here with our GW-PPO variant).

The findings have potential implications in cognitive neuroscience as a practical test of GWT. First, compared with existing approaches (Radford et al., 2021; Hafner et al., 2024), the GW tends to produce a superior multimodal representation, owing to its semi-supervised training procedure inspired by the "broadcast" principle (Baars, 1988). This fact was already suggested by a number of recent studies (Devillers et al., 2021, 2024; Maytié et al., 2024), and is confirmed here in the context of model-based RL. Second, we show that GW multimodal representations can be leveraged by a World Model to produce mental simulations that help the system converge to an optimal decision strategy. This resembles "dreaming" in humans and animals, and more generally, captures the ability of such biological systems to imagine the potential outcome of a planned sequence of actions before making a decision.

While this evidently does not suffice to entirely validate GWT, it confirms its potential relevance as a theory of higher-level cognition.

Ultimately, this research tackles key challenges in RL, such as the large amount of environment interactions required for policy training and the need for strong multimodal representations, particularly in robotics. It opens the way for future work to further integrate GWT and World Models.

# **Acknowledgments**

This work was supported by an ANITI Chair (ANR grant ANR-19-PI3A-004), an ANR grant COCOBOT (ANR21-FAI2-0005) and by "Défi Clé Robotique centrée sur l'humain" funded by Région Occitanie, France. This research is also part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No.101096017).

# References

- Atkeson, C., & Santamaria, J. (1997). A comparison of direct and model-based reinforcement learning. In *Proceedings of International Conference on Robotics and Automation* (Vol. 4). Retrieved from https://ieeexplore.ieee.org/document/606886 doi: 10.1109/ROBOT.1997.606886
- Baars, B. J. (1988). *A Cognitive Theory of Consciousness*. New York: Cambridge University Press.
- Bruce, J., Dennis, M., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., ... Rocktäschel, T. (2024). *Genie: Generative Interactive Environments*. arXiv. Retrieved from http://arxiv.org/abs/2402.15391 (arXiv:2402.15391 [cs])
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734). Doha, Qatar: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D14-1179/ doi: 10.3115/v1/D14-1179
- Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *The Behavioral and Brain Sciences*, *36*(3), 181–204. doi: 10.1017/S0140525X12000477
- Collaboration, O. X.-E., O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., ... Lin, Z. (2024). Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration0. In 2024 ieee international conference on robotics and automation (icra). Retrieved from http://arxiv.org/abs/2310.08864 doi: 10.1109/ICRA57147.2024.10611477
- Dearden, R., Friedman, N., & Russell, S. (n.d.). Bayesian Q-Learning.

- Dehaene, S., Kerszberg, M., & Changeux, J.-P. (1998). A neuronal model of a global workspace in effortful cognitive tasks. *Proceedings of the National Academy of Sciences*, 95(24), 14529–14534. Retrieved from https://www.pnas.org/doi/full/10.1073/pnas.95.24.14529 doi: 10.1073/pnas.95.24.14529
- Devillers, B., Choksi, B., Bielawski, R., & VanRullen, R. (2021). Does language help generalization in vision models? In *Proceedings of the 25th Conference on Computational Natural Language Learning*. Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.conll-1.13/ doi: 10.18653/v1/2021.conll-1.13
- Devillers, B., Maytié, L., & VanRullen, R. (2024). Semi-Supervised Multimodal Representation Learning Through a Global Workspace. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15. Retrieved from https://ieeexplore.ieee.org/abstract/document/10580966 (Conference Name: IEEE Transactions on Neural Networks and Learning Systems) doi: 10.1109/TNNLS.2024.3416701
- Friston, K. (2010). The free-energy principle: a unified brain theory? Nature Reviews Neuroscience, 11(2), 127–138. Retrieved from https://www.nature.com/articles/nrn2787 doi: 10.1038/nrn2787
- Ha, D., & Schmidhuber, J. (2018). Recurrent World Models Facilitate Policy Evolution. In Advances in Neural Information Processing Systems (Vol. 31). Curran Associates, Inc. Retrieved from https://papers.nips.cc/paper\_files/paper/2018/ hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html
- Hafner, D., Pasukonis, J., Ba, J., & Lillicrap, T. (2024). Mastering diverse domains through world models. Retrieved from https://arxiv.org/abs/2301.04104
- Konda, V., & Tsitsiklis, J. (1999). Actor-Critic Algorithms. In Advances in Neural Information Processing Systems (Vol. 12). MIT Press. Retrieved from https://papers.nips.cc/paper\_files/paper/1999/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html
- Maytié, L., Devillers, B., Arnold, A., & VanRullen, R. (2024). Zero-shot cross-modal transfer of Reinforcement Learning policies through a Global Workspace. *Reinforcement Learning Journal*, 3. Retrieved from http://arxiv.org/abs/2403.04588 (ISSN: 2996-8577 arXiv:2403.04588 [cs]) doi: https://doi.org/10.48550/arXiv.2403.04588
- NVIDIA, Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., ... Zolkowski, A. (2025). *Cosmos World Foundation Model Platform for Physical Al.* arXiv. Retrieved from http://arxiv.org/abs/2501.03575 (arXiv:2501.03575 [cs]) doi: 10.48550/arXiv.2501.03575
- Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., . . . Bojanowski, P. (2023). DINOv2: Learn-

- ing Robust Visual Features without Supervision. *Transactions on Machine Learning Research*. Retrieved from https://openreview.net/forum?id=a68SUt6zFt
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. arXiv. Retrieved from http://arxiv.org/abs/2103.00020 (arXiv:2103.00020 [cs]) doi: 10.48550/arXiv.2103.00020
- Safron, A. (2022). Integrated world modeling theory expanded: Implications for the future of consciousness. Frontiers in Computational Neuroscience, 16. Retrieved from https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2022.642397/full doi: 10.3389/fncom.2022.642397
- Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.*, 2(4), 160–163. Retrieved from https://dl.acm.org/doi/10.1145/122344.122377 doi: 10.1145/122344.122377
- Szita, I., & Szepesvári, C. (n.d.). Model-based reinforcement learning with nearly tight exploration complexity bounds.
- VanRullen, R., & Kanai, R. (2021). Deep learning and the Global Workspace Theory. *Trends in Neurosciences*, 44(9), 692–704. Retrieved from https://www.cell.com/trends/neurosciences/abstract/S0166-2236(21)00077-1 doi: 10.1016/j.tins.2021.04.005
- Walke, H. R., Black, K., Zhao, T. Z., Vuong, Q., Zheng, C., Hansen-Estruch, P., ... Levine, S. (2023). BridgeData V2: A Dataset for Robot Learning at Scale.. Retrieved from https://openreview.net/forum?id=f55MlAT1Lu

## **Model Parameters**

#### **GW Parameters**

This section details the architecture used in the Global Workspace. It begins with the VAEs used to pass from raw observations  $(o^v, o^{attr})$  to latent unimodal representation  $(z^v, z^{attr})$ . The visual VAE, detailed in Table 1, is composed of Convolutional Layers with Batch Norm and ReLU. Its latent dimension is of size 10 and it counts 5.8M parameters in total. The attributes VAE, detailed in Table 2, is much smaller, with 11,000 parameters. It is composed of Multiple Linear and ReLU layers with a latent dimension of 10. The last layer of the decoder is divided in two parts, one of size 3 to predict the class of the shape (one-hot encoding), another one of size 8 with a Tanh activation to predict the other attributes. These VAEs are used for the Global Workspace model, but also for VAE-PPO and VAE-Dreamer.

VAE encoder $(2.8M \text{ params})$	VAE decoder (3M params)
$x \in \mathbb{R}^{3 \times 32 \times 32}$	$z \in \mathbb{R}^{10}$
$Conv_{128} - BN - ReLU$	FC <sub>8×8×1024</sub>
$Conv_{256} - BN - ReLU$	$ConvT_{512} - BN - ReLU$
$Conv_{512} - BN - ReLU$	$ConvT_{256} - BN - ReLU$
$Conv_{1024} - BN - ReLU$	$ConvT_{128} - BN - ReLU$
$Flatten - FC_{2 \times 10}$	Conv <sub>1</sub> — Sigmoid

Table 1: Architecture and number of parameters of the visual VAE used to encode  $o^{v}$  to  $z^{v}$ .

VAE encoder $(6,700 \text{ params})$	VAE decoder (4,700 params)
$x \in \mathbb{R}^{11}$	$z \in \mathbb{R}^{10}$
FC <sub>64</sub> — ReLU	$\begin{aligned} & FC_{64} - ReLU \\ & FC_{64} - ReLU \\ & FC_{3} \times FC_{8} - Tanh \end{aligned}$
FC <sub>64</sub> — ReLU	FC <sub>64</sub> — ReLU
$FC_{10}$ — ReLU	$FC_3 \times FC_8 - Tanh$
$FC_{2\times 10}$	

Table 2: Architecture and number of parameters of the attributes VAE used to encode  $o^{attr}$  to  $z^{attr}$ .

Table 3 gives details about encoders  $(e_v, e_{attr})$  and decoders  $(d_v, d_{attr})$  architecture of the Global Workspace. Because they are identical for both modalities only one table provides the architecture's details. The encoders and decoders are simply a sequence of Linear layers with ReLU activation function, and the latent dimension of the GW is of size 10. This GW model takes inputs from the VAEs described before and is used in the GW-Dreamer and GW-PPO models.

# **WM Parameters**

This part gives details about the architectures and parameters used for the World Model inside GW-Dreamer. It is composed of two main elements. First, a dynamic part, which is a one-layer GRU counting 16,000 parameters. It takes as input a vector of size 17 (GW representation and actions) with a hidden representation  $h_t$  of size 64. Second, it has 3 different heads to retrieve different information from the GRU

GW encoder (13,800 params)	GW decoder (13,800 params)
$x \in \mathbb{R}^{10}$	$z \in \mathbb{R}^{10}$
FC <sub>64</sub> — ReLU	FC <sub>64</sub> — ReLU
FC <sub>64</sub> — ReLU	FC <sub>64</sub> — ReLU
FC <sub>64</sub> — ReLU	$\begin{aligned} &FC_{64} - ReLU \\ &FC_{64} - ReLU \\ &FC_{64} - ReLU \\ &FC_{64} - ReLU \end{aligned}$
FC <sub>64</sub> — ReLU	FC <sub>64</sub> — ReLU
FC <sub>10</sub>	FC <sub>10</sub>

Table 3: Architecture and number of parameters of the encoders and decoders of the Global Workspace model for one modality.

latent space. The architecture of the different heads is detailed in Table 4. The head used to predict the GW latent vector is a simple linear layer with a Tanh activation function. The two others (predicting the scalar reward and a Boolean "done" termination flag) have an identical architecture. They are composed of a sequence of Linear layers and ReLU activation function. These heads count 429,000 parameters, and in total the World Model is composed of 445,000 parameters

GW head (650 params)	reward / done head (214,000 params)
$x \in \mathbb{R}^{64}$	$z \in \mathbb{R}^{64}$
$FC_{10}$ — Tanh	$\begin{aligned} &FC_{256} - ReLU \\ &FC_{256} - ReLU \\ &FC_{256} - ReLU \end{aligned}$
	FC <sub>256</sub> – ReLU
	FC <sub>256</sub> – ReLU
	FC <sub>1</sub>

Table 4: Architecture and number of parameters of the different heads composing the World Model.

#### **AC Parameters**

Finally, this part describes the number of parameters used for the Actor-Critic inside the GW-Dreamer model. The architecture is similar to the one proposed in Dreamer (Hafner et al., 2024). As shown in Table 5 they are composed of a sequence of Linear layers with ReLU activation function. The Actor outputs the parameters of a distribution (mean, variance) about possible actions. The Critic also predicts a distribution about the value of a state. As in Dreamer, the space is discretized in different bins to apply a categorical cross-entropy loss function between the two hot-encoded targets and the predicted softmax distribution (for more details, see Dreamer implementation (Hafner et al., 2024)).

Actor (500K params)	
$x \in \mathbb{R}^{64}$	$z \in \mathbb{R}^{64}$
$FC_{512}$ — ReLU	FC <sub>512</sub> – ReLU FC <sub>512</sub> – ReLU
$FC_{512}$ — ReLU	FC <sub>512</sub> – ReLU
$FC_{2\times7}$	$FC_{2 \times 255}$

Table 5: Architecture and number of parameters of the Actor-Critic.

# **GW losses Parameters**

This section gives more details about the hyperparameters used during the training of the GW model. The losses used to train the model are described in Equations 2 and 3. These losses are scaled by different weights that we fixed as follows:

$$\begin{cases} \beta dcy = 1\\ \beta cy = 1\\ \beta tr = 1\\ \beta fusion = 1\\ \beta cont = 0.1 \end{cases} \tag{4}$$

These values were taken from the implementation done by Devillers et al. (2024), where the weight of the contrastive loss (measured by a cross-entropy function) was always smaller compared to the other ones (measured using MSE distance).