Adversarial Curriculum Generation for World Models in Reinforcement Learning

Siyao Li
King's College London
London, United Kingdom
siyao.2.li@kcl.ac.uk

Matteo Leonetti
King's College London
London, United Kingdom
matteo.leonetti@kcl.ac.uk

ABSTRACT

World models enable efficient policy learning by replacing costly environment interactions with planning and imagination in latent space. However, world models are often trained on data collected from fixed or randomly generated environments. We introduce Model-driven Adversarial Curriculum (MAC), a closed-loop approach that actively generates training environments for world model learning. MAC couples procedural generation with an adversarial editor that is trained to generate environments with high world model prediction error, thereby exposing underrepresented and poorly predicted transitions. This enables a single world model to continually accumulate reusable dynamics knowledge across a curriculum of training environments.

To support transfer across diverse environments, we further propose an Agent-Centric Attentive World Model (AC-AWM) that uses local, action-conditioned attention to disentangle agent-centric dynamics from background variation. Experiments on MiniGrid show that MAC improves zero-shot generalization to unseen target tasks and is substantially more sample-efficient than training on randomly generated environments and direct training on target tasks.

KEYWORDS

Curriculum learning, Model-based reinforcement learning, Unsupervised environment design

1 INTRODUCTION

Learning effective world models is increasingly viewed as a key challenge for autonomous agents, since predictive models enable planning and imagination in latent space and can substantially improve data efficiency in reinforcement learning [9, 13, 20]. Unlike policy learning, which focuses on acquiring task-specific behaviors, world models aim to capture environment dynamics, which transfer across tasks and settings [7, 24]. Consequently, the distribution of training experience directly determines which dynamics are modeled and which remain underrepresented. Despite this central role of experience, most world model learning pipelines still rely on collecting large amounts of interaction data from fixed simulators [3, 11]. Alternatively, training may be performed across randomly varied environments, which involves generating multiple environment instances [21]. Such training regimes are often inefficient and may fail to expose rare but highly informative interactions, delaying the emergence of accurate and reliable predictive representations. These limitations suggest that the bottleneck lies not only in model capacity, but in how training environments are structured.

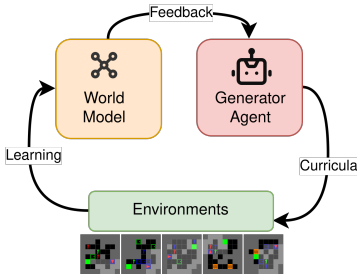

Figure 1: Model based curriculum learning. A generator creates training tasks guided by feedback from the world model, establishing a closed-loop curriculum learning process.

From this perspective, training environments can be viewed as generative processes over environment configurations—effectively parameterized "worlds" [17, 18]. If the environment distribution governs the transition distribution observed by the agent, then shaping this distribution becomes a principled mechanism for accelerating world model learning. Curriculum learning (CL), which organizes experience through structured progression, naturally emerges as a candidate solution.

CL has long been central to improving policy learning, progressing from manually designed curricula to automated and adversarial schemes that adapt task difficulty to agent competence [1, 5, 23]. Recent work on Unsupervised Environment Design (UED) similarly aims to learn generalized policies, which perform robustly across diverse environment instances rather than overfitting to a single configuration, by generating training environments that induce learning progress and robust behavior [4]. However, these approaches primarily target policy optimization rather than systematic modeling of environment dynamics.

Consequently, curriculum mechanisms specifically designed for world model training remain comparatively underdeveloped and are often applied in a manual or ad-hoc manner, for example through hand-crafted task suites or fixed domain randomization. While maximizing environment diversity increases variation, diversity alone does not guarantee exposure to transitions that are underrepresented or poorly predicted by the current model. Without guided progression, a world model may waste capacity modeling trivial transitions or fail to discover rare, complex interactions essential for robust prediction. Effective learning therefore requires not exhaustive variation, but targeted exposure aligned with the model's predictive competence and learning progress [10, 16, 19].

Proc. of the Adaptive and Learning Agents Workshop (ALA 2026), Aydeniz, Delgrange, Mohammedalamen, Yang (eds.), May 25 – 26, 2026, Paphos, Cyprus, https://alaworkshop2026.github.io/. 2026.

A fundamental gap remains in the current landscape: to the best of our knowledge, principled curriculum mechanisms explicitly developed for training generalized world models remain largely unexplored, while curriculum mechanisms are well-established for policy optimization. To this end, we propose the Model-driven Adversarial Curriculum (MAC) framework. Central to MAC is an Agent-Centric Attentive World Model (AC-AWM), designed to decouple agent ego-motion from varying background features. This architecture enables knowledge reuse across heterogeneous environments, allowing the curriculum to efficiently target the domain's underlying dynamics. As illustrated in Figure 1, MAC operates as a closed-loop process where a Procedural Content Generation (PCG) component supplies diverse canvases, and an editor agent performs targeted modifications based on the model's predictive error. By focusing on underrepresented transitions, MAC ensures the rapid learning of a robust and generalized world model.

2 RELATED WORK

2.1 Model Based Reinforcement Learning

Model-based reinforcement learning (MBRL) improves decision-making by learning an explicit predictive model of environment dynamics and using it for planning or policy optimization. By enabling agents to perform imagined rollouts within the learned model, MBRL can substantially improve sample efficiency, trading expensive real-environment interactions for comparatively cheap model-based simulations. Early work demonstrated that learning latent dynamics from high-dimensional observations can enable control directly from pixels [24]. Recent MBRL methods learn compact latent world models and perform planning or imagination-based policy learning in the latent space, e.g., PlaNet [7] and Dreamer [6, 8, 9] . Complementary lines of work study how to leverage learned models efficiently for policy improvement, including uncertainty-aware planning (e.g., PETS [3]) and limiting model rollout horizons to mitigate compounding error (e.g., MBPO [11]). Large-scale results further show that planning with learned models can drive strong performance in complex domains, such as MuZero in Atari and board games [20]. Related to our attention-based design, Zhao et al. introduce a planning agent with a set-based representation and an attention bottleneck to promote generalization across environment variations [26]. Like their work, we employ attention to support transferable representations across diverse environments. However, while their approach relies on object-centric set encodings, our AC-AWM operates directly on raw spatial patches with early action fusion, enabling action-driven dynamics learning without explicit object extraction.

While these methods advance model learning and model utilization, they typically assume a fixed data-collection regime (e.g., interaction in a given simulator or a fixed task suite) and do not directly address how to design a curriculum of environments that can accelerate world model learning and support the accumulation of reusable dynamics knowledge in a single continually trained model across a sequence of training environments.

2.2 Curriculum Learning and Unsupervised Environment Design

Curriculum learning (CL) strategies organize training experiences from simple to complex to accelerate convergence and improve generalization [1]. As detailed in comprehensive surveys [15, 17], the application of CL to reinforcement learning has evolved distinctively. Early approaches primarily relied on *hand-crafted* curricula, where task sequences or difficulty schedules were manually defined based on human domain expertise [25]. While effective in controlled settings, such manually defined curricula are inherently limited in scalability, as they require extensive domain knowledge and do not adapt to the agent's evolving competence.

To overcome the scalability limits of manual design, research shifted toward *Automatic Curriculum Learning* (ACL), where the learning process itself dictates the progression of tasks. Representative methods include reverse curriculum generation, which adapts the start-state distribution based on the agent's current competence [5]. Most recently, this paradigm has expanded into *Unsupervised Environment Design* (UED), which formalizes the curriculum not just as a selection of tasks, but as a generative process over the environment's structure and physics. Unlike standard domain randomization, open-ended approaches such as POET [23] and PAIRED [4]co-evolve environments alongside the agent, providing a principled framework for exposing under-trained behaviors.

Crucially, such curriculum generation and UED methods are primarily designed to improve policy learning, typically in model-free reinforcement learning settings.

In contrast, our work focuses on how curriculum learning can be leveraged to efficiently train a generalized world model. We propose a model-driven curriculum mechanism that shapes the training data distribution through environment generation, explicitly targeting underrepresented state–action–state transitions to accelerate robust dynamics acquisition. To enable effective knowledge transfer across diverse environment structures, we further design a transferable agent-centric world model architecture that supports cross-environment generalization.

3 PRELIMINARIES

We consider a model-based reinforcement learning setting formulated as a fully observable Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, \mathcal{R}, \gamma \rangle$. We assume the reward function $\mathcal{R}$ is known and focus on learning a parametric world model f_θ to approximate the transition dynamics $P(s_{t+1}|s_t, a_t)$. The model is trained via supervised learning on transition data $\mathcal{D} = \{(s_t, a_t, s_{t+1})\}$ to minimize the prediction error:

$$\mathcal{L}_{\text{WM}}(\theta) = \mathbb{E}_{\mathcal{D}} \left[\|s_{t+1} - \hat{s}_{t+1}\|_2^2 \right]. \tag{1}$$

3.1 Egocentric Residual World Model

To enable generalization across structurally distinct environments, we incorporate structural inductive biases that decouple local interaction rules from global layout variations. We propose an **Egocentric Residual** formulation (Fig. 2) that we will define using our MiniGrid domain as an illustrative example (Fig. 3):

Local Observation (Φ). Instead of processing the full global state s_t, we define a masking operator $\Phi : \mathcal{S} \to \mathcal{S}_{loc}$ to extract an

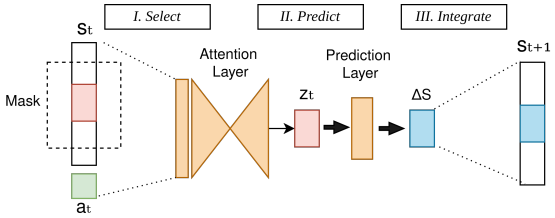

Figure 2: Architecture of the Agent-Centric Attentive World Model (AC-AWM). The masked agent-centered state is encoded by convolutional layers, combined with an action embedding, and processed by a Transformer encoder to predict residual dynamics.

agent-centered local observation $o_t = \Phi(s_t)$. The local observation must be sufficient to predict the consequences of the actions, as expressed by the residual dynamics. *In the context of MiniGrid*, we implemented the observation function by extracting a local window (e.g., 7×7) around the agent's current position. The local window ensures that the model input captures relevant local objects (walls, keys, doors) while remaining invariant to the agent's absolute coordinates in the global map. The effects of the actions never depend on objects outside of this local window. In this work we defined the observation function manually, but we envision that it can also be learned.

Residual Dynamics. The model predicts the incremental change in state rather than the absolute next state [14]:

$$\hat{s}_{t+1} = s_t + f_\theta(o_t, a_t). \tag{2}$$

This residual formulation biases the model toward learning local physical rules. For example, in MiniGrid, when the agent executes a "move forward" action, the model does not predict an entirely new global grid. Instead, it predicts local changes: the current cell transitions from `agent` to `empty`, while the forward cell transitions from `empty` to `agent`. These localized updates are then added to the global state via the residual mechanism, enabling the learned interaction rules to generalize across different layouts.

3.2 Model-Driven Curriculum Learning

We employ a feedback-driven curriculum to structure the training process. Our approach dynamically adapts the task distribution $p(\mathcal{T})$ based on the world model's validation performance, rather than follow a fixed progression. Specifically, the generator constructs environments that induce high prediction error (high epistemic uncertainty) in f_θ. This active selection mechanism focuses training resources on the model's current weaknesses, ensuring efficient coverage of the dynamics space.

4 METHODOLOGY

4.1 Agent-Centric Attentive World Model

Figure 2 provides an overview of the proposed Agent-Centric Attentive World Model (AC-AWM). The model extracts a masked agent-centered observation, encodes it through convolutional layers, fuses action information via a Transformer encoder, and predicts residual dynamics that are integrated into the global state.

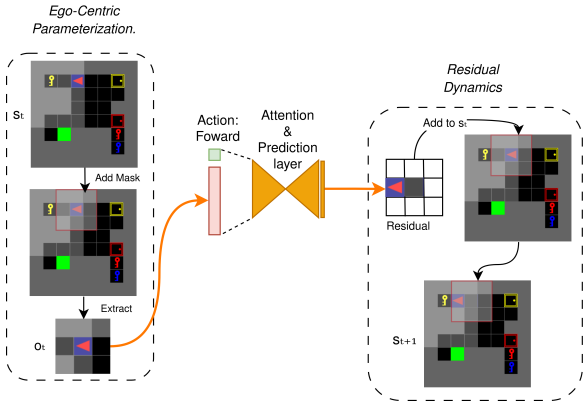

Figure 3: AC-AWM Architecture illustrated on a MiniGrid example. Left: Extraction of egocentric observation o_t from global state s_t. Right: Residual dynamics integration, where a predicted local residual is added to s_t to yield $\hat{s}_{t+1}$.

As described in Section 3, we parameterize an egocentric residual transition model $f_\theta(o_t, a_t)$ using this attention-based architecture. We assume that the global state admits a tensor representation $s_t \in \mathbb{R}^{H \times W \times c}$, where H and W denote the spatial dimensions of the full environment, and c denotes the feature channel dimension, encoding attributes such as object type, color, or object status (e.g., closed or opened doors).

To introduce an inductive bias of action locality, we define a masking operator $\phi : \mathcal{S} \to \mathcal{S}_{loc}$, which extracts a local agent-centered window from the global state. The resulting observation is $o_t = \phi(s_t)$, with $o_t \in \mathbb{R}^{h \times w \times c}$, where h and w denote the spatial dimensions of the local window.

Tokenization. The masked observation o_t serves as the input to the dynamics model. The input is processed by convolutional layers, producing feature maps $F_t \in \mathbb{R}^{h' \times w' \times d}$.where h' and w' denote the spatial dimensions after convolutional downsampling. The spatial map is reshaped into $N = h'w'$ tokens:

$$X_t \in \mathbb{R}^{N \times d}.$$

Learnable positional encodings $P \in \mathbb{R}^{N \times d}$ are added:

$$\tilde{X}_t = X_t + P.$$

Action-Conditioned Transformer. The action a_t is embedded as $e_t \in \mathbb{R}^d$ and appended as an additional token:

$$\bar{X}_t = [e_t; \tilde{X}_t].$$

The sequence is processed by a Transformer encoder [22]:

$$Z_t = \text{Transformer}(\bar{X}_t).$$

The output corresponding to the action token serves as latent state $z_t \in \mathbb{R}^d$.

Residual Prediction. Following the residual dynamics formulation introduced in Section 3, the multilayer perceptron predicts a residual tensor

$$f_\theta(z_t) \in \mathbb{R}^{H \times W \times c},$$

Algorithm 1: MAC: Model-driven Adversarial Curriculum

Input: Initial World Model θ_0, Editor π_ϕ, Encoder f_ω, Generator G_{base}, Buffer $\mathcal{B}$
Output: Robust World Model θ^*

Initialize θ, ϕ, ω; Initialize Fisher Info F and Old Params $\theta_{\text{old}} \leftarrow \theta_0$;
for *iteration* $t = 1, \ldots, T$ **do**
  `// Phase 1: Structured Adversarial Generation`
  Generate $\{\psi_{\text{base},i}\} \leftarrow G_{\text{base}}$;
  foreach *configuration* $\psi \in \{\psi_{base,i}\}$ **do**
    $c \leftarrow f_\omega(\psi, \mathcal{E}_{t-1})$;
    $a_e \sim \pi_\phi(\cdot \mid \psi, c)$;
    $\psi' \leftarrow \Gamma(\psi, a_e)$;
    Collect trajectory τ in ψ' with random policy;
    if ψ' *is Valid* **then**
      $\mathcal{D}_{\text{curr}} \leftarrow \mathcal{D}_{\text{curr}} \cup \{\tau\}$;
      $r \leftarrow \lambda_{\text{err}} \mathcal{L}_{\text{WM}}(\tau; \theta) + \lambda_{\text{val}} R_{\text{val}} + \lambda_{\text{div}} R_{\text{div}}$;
    end
    else
      $r \leftarrow -P_{\text{fail}}$;
    end
  end
  `// Phase 2: Update Generator and World Model`
  Update ϕ, ω using PPO;
  $\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{curr}} \cup \text{Sample}(\mathcal{B})$;
  Update θ using EWC loss;
  Update $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{D}_{\text{curr}}$;
  `// Curriculum-level convergence`
  if $|\mathcal{L}_{WM}^{(t)} - \mathcal{L}_{WM}^{(t-1)}| < \epsilon$ **then**
    break;
  end
end

which is non-zero only over the masked local region defined by $\phi(s_t)$ and zero elsewhere. The next-state prediction is obtained via residual update:

$$\hat{s}_{t+1} = s_t + f_\theta(z_t).$$

Training. The model is trained using supervised one-step residual prediction. Given transition tuples (s_t, a_t, s_{t+1}), we compute the ground-truth residual

$$r_t^* = s_{t+1} - s_t,$$

and minimize

$$\mathcal{L}_{\text{WM}}(\theta) = \mathbb{E}\left[\|f_\theta(z_t) - r_t^*\|_2^2\right].$$

4.2 Model-driven Adversarial Curriculum

We build upon the hierarchical formulation introduced in Unsupervised Environment Design (UED) [4], which models environment generation as a Configuration MDP coupled with an agent-level

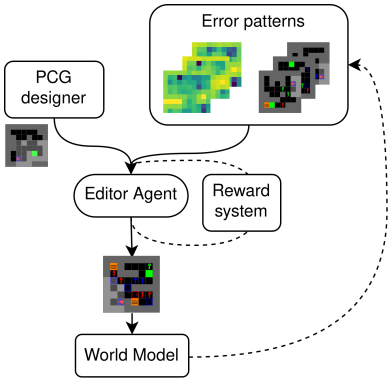

Figure 4: Model-driven adversarial curriculum framework. The editor modifies an initial configuration ψ_{base} to maximize world model prediction error while maintaining task validity.

Environment MDP. Unlike UED, which optimizes environment generation to improve policy robustness, we adapt this hierarchical structure for world model learning. Specifically, the editor is driven by world model prediction error rather than policy regret, enabling targeted exposure to underrepresented dynamics. The overall workflow is summarized in Algorithm 1. We propose a model-driven curriculum framework for world model learning as shown in figure 4. The framework treats environment design as a hierarchical process involving two distinct processes: an Environment MDP $\mathcal{M}_{\text{env}}(\psi)$, where an agent interacts with states $s \in \mathcal{S}$ to generate transitions, and a Configuration MDP $\mathcal{M}_{\text{edit}}$, where an adversarial editor modifies the parameterization $\psi \in \Psi$ based on model performance. The overall workflow is summarized in Algorithm 1.

4.2.1 Procedural Generation of Minimal Environments. The curriculum begins with an initial configuration ψ_{base} sampled from a procedural content generator (PCG). This minimal structural configuration defines the underlying spatial or physical layout of the environment before task-specific elements are introduced. Such initialization ensures that environments start from a structurally valid region of the configuration space Ψ, upon which adversarial modifications are subsequently applied by the editor.

4.2.2 Adversarial Editor MDP. The editing process is formulated as a Configuration MDP $\langle \mathcal{S}_e, \mathcal{A}_e, \mathcal{P}_e, \mathcal{R}_e \rangle$ that refines the initial structural configuration to maximize world model learning progress.

Observation and Action Space. The editor state s_e is an augmented observation combining structural configuration and epistemic feedback. It consists of the current configuration $M \in \Psi$ and a latent context c. The context c is produced by a history encoder $f_\omega(M, \mathcal{E})$ that aggregates previous prediction errors $\mathcal{E}$ to identify undermodeled dynamics. The editor policy is defined as $\pi_\phi(a_e \mid M, c)$. The history embedding represents the prediction error pattern from the previous curriculum iteration. It does not encode full trajectory history and no recurrent hidden state is maintained.

Unlike incremental editing schemes, the editor produces a structured action a_e in a single decision step, which specifies a complete modification of the base configuration. To ensure controlled modifications, we impose a fixed edit ratio ρ, meaning that at most a

fraction ρ of the editable grid cells can be modified. The resulting environment is obtained by applying this structured action to the base configuration:

$$\psi' = \Gamma(M, a_e),$$

where Γ denotes a deterministic application operator. The editor action a_e parameterizes incremental modifications to a base environment configuration, rather than specifying a complete environment from scratch.

In our MiniGrid instantiation, s_e consists of a base grid layout with only empty cells and walls, and a latent error pattern encoding per-cell prediction errors from the previous curriculum iteration. The editor action a_e is a structured modification matrix, where each cell specifies a local change (e.g., inserting a colored key, door, lava, or no-op), which is applied to the base layout to generate ψ'.

Curriculum-Level Convergence. Since editing is performed in a single structured update, termination is not defined over intermediate editing steps but at the curriculum level. We stop curriculum expansion when newly generated and edited environments no longer induce meaningful changes in the world model feedback signal. Let ψ'_t denote the edited environment generated at iteration t, and let $\text{Eval}(\psi'_t, \theta)$ be a scalar feedback signal (e.g., prediction loss) computed from trajectories collected in ψ'_t. We declare convergence when:

$$\left|\text{Eval}(\psi'_t, \theta) - \text{Eval}(\psi'_{t-1}, \theta)\right| < \epsilon.$$

This indicates saturation of learning progress: the generated tasks are no longer producing substantial new error patterns, suggesting that the world model has largely captured the dominant dynamics exposed by the curriculum.

4.2.3 Curriculum Reward Formulation. To assess a generated configuration ψ', we define a terminal reward $R(\psi')$ based on the dynamics induced by the configuration. While the overall structure follows prior generative curriculum frameworks that evaluate environments at the trajectory level, our reward formulation is explicitly driven by world model prediction behavior rather than policy regret or task difficulty. We define the reward on a trajectory rather than on individual editor actions because the world model's learning progress is an emergent property of the environment's global dynamics.

Trajectory Evaluation. We evaluate ψ' by collecting a trajectory τ of horizon H sampled from $\mathcal{M}_{\text{env}}(\psi')$ using a random exploration policy π_{rand}:

$$\tau = \{(s_t, a_t, s_{t+1}) \mid s_0 \sim \mu_0(\psi'), a_t \sim \pi_{\text{rand}}, s_{t+1} \sim \mathcal{P}(\psi')\}_{t=0}^{H-1}. \quad (3)$$

We use random exploration for trajectory collection. As our goal is dynamics modeling rather than policy learning, training an exploration policy would add unnecessary computational cost.

Reward Components. The total reward is a weighted sum of three complementary terms:

$$R(\psi') = \lambda_{\text{err}} R_{\text{error}}(\tau) + \lambda_{\text{val}} R_{\text{validity}}(\psi') + \lambda_{\text{div}} R_{\text{diversity}}(\psi'). \quad (4)$$

Adversarial Error Reward. The term R_{error} encourages configurations that expose under-modeled dynamics. It is defined as the Mean Squared Error (MSE) of the world model:

$$\mathcal{L}_{\text{WM}}(\tau; \theta) = \frac{1}{H} \sum_{t=0}^{H-1} \|s_{t+1} - \hat{s}_{t+1}\|^2.$$

Higher prediction error indicates regions where the world model lacks accuracy, guiding the generator toward challenging structural regimes.

Validity Reward. The term R_{validity} ensures that generated environments permit sufficient state-space traversal. Unsolvable configurations may confine the agent to restricted regions, limiting exposure to critical dynamics near structural bottlenecks or goal-related areas. By enforcing solvability (e.g., via a BFS-based solver), we promote broader transition coverage and improve data efficiency for world model learning.

Diversity Reward. To prevent curriculum collapse, $R_{\text{diversity}}$ rewards structurally novel configurations in a learned embedding space. We first obtain a representation $z_\psi = f_{\text{target}}(\psi)$ using the fixed target network of Random Network Distillation (RND [2]). Novelty is then measured as the k-nearest neighbor (k-NN) distance between z_ψ and previously generated configurations stored in an archive. Configurations that are far from existing embeddings receive higher diversity reward, encouraging exploration of distinct environment configurations.

4.2.4 World Model Optimization. As the editor shifts the environment distribution, the world model θ must adapt without catastrophic forgetting. We employ Elastic Weight Consolidation (EWC [12]) to stabilize this non-stationary learning process. The total loss for θ is:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{WM}}(\mathcal{D}_{\text{train}}; \theta) + \frac{\lambda_{\text{ewc}}}{2} \sum_i F_i(\theta_i - \theta_{\text{old},i})^2, \quad (5)$$

where F is the Fisher Information Matrix and θ_{old} represents the parameters from the previous curriculum iteration.

4.2.5 Implementation Details in Gridworlds. We instantiate this framework in the MiniGrid domain, where Ψ represents $H \times W$ grids. Epistemic feedback $\mathcal{E}$ is implemented as a spatial heatmap of prediction errors processed by a CNN history encoder.

Editor actions a_e consist of cell-type toggling atop the initial PCG scaffold. Feasibility is verified via BFS, and unsolvable configurations are penalized by P_{fail}, while solvable tasks receive a complexity bonus proportional to the shortest path length L_{bfs}.

5 EXPERIMENTS

5.1 Minigrid Environment and Implementation Details

We evaluate our framework on the MiniGrid domain, a grid-world environment that challenges agents with fully observable navigation and object interaction tasks. The environment consists of various objects including walls, doors, keys, and hazards (lava), where the agent receives a compact vector observation encoding object types, colors, and states. Note that for this study, we focus on the prediction of immediate physical interactions and diverse object dynamics rather than long-horizon sequential dependencies. Therefore, we exclude the "locked door" mechanic—which necessitates recurrent memory for inventory tracking—and instead

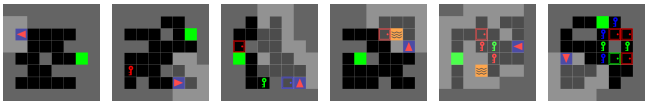

Figure 5: MAC-generated minitasks from early training iterations to later stages.

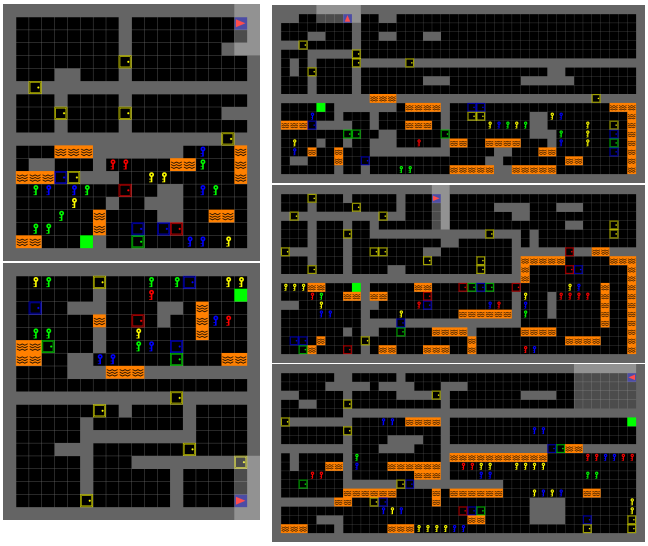

Figure 6: Target Tasks used for evaluation.

prioritize maximizing the combinatorial diversity of layouts and element interactions. Despite this modification, due to its combinatorial complexity and sparse rewards, this setup serves as an ideal testbed for evaluating the predictive dynamics modeling capabilities of world models. For this paper, experiments are conducted in a deterministic version of the environment. Extending the framework to stochastic dynamics is left for future work.

5.1.1 Minitask Design. To bootstrap the learning process, our generator creates "minitasks" on a restricted 8×8 canvas. These minitasks are procedurally initialized and then adversarially edited to serve as "atomic" learning units. The small scale allows for dense feedback and rapid iteration, while still capturing the fundamental physics of the larger domain, as illustrated in Fig. 5.

Validation & Target Task Set. To explicitly evaluate the out-of-distribution (OOD) and zero-shot generalization capability of the learned world model, we construct a fixed set of 15 challenging Target Tasks (Fig. 6) that are *never encountered during training*. These tasks exhibit complex layouts and specific mechanical interactions (e.g., reaching a goal behind multiple closed doors), which are unlikely to be fully covered by random exploration or curriculum-generated minitasks.

In addition, to provide a ground-truth reference for evaluation, we collect a Uniform Exploration Dataset from these target environments, ensuring broad coverage of the underlying state space for baseline comparison.

5.1.2 Model Configurations. The World Model is a AC-AWM model implemented as an attention-based dynamics model that predicts future states. It employs a local attention mechanism with a 3×3 mask to capture spatial dependencies within the 8×8 grid. The model uses an embedding dimension of 256 with a single attention head for computational efficiency. To mitigate catastrophic forgetting during curriculum learning, we incorporate a Fisher-weighted replay buffer with a capacity of 500,000 transitions.

The Editor Agent is a PPO-based policy with a global context dimension of 64 and a history embedding dimension of 16. It operates on batches of 6 minitasks per iteration.

5.1.3 Weighted Loss Optimization. To address the severe class imbalance between static background transitions and sparse object interactions, we employ an interaction-aware weighted loss mechanism. Specifically, we assign significantly higher weights to pixels exhibiting state changes ($w = 100$) or agent-object interactions, compared to static background regions ($w = 1$). This weighting is critical for validating the effectiveness of our curriculum: without it, the accuracy improvements on complex interactions generated by the curriculum would be masked by the vast dominance of static background data, making it impossible to distinguish a robust world model from a trivial baseline.

5.1.4 Reproducibility. The code will be released upon acceptance.[1] All experiments are conducted with 5 independent random seeds. We report the mean performance across seeds for all quantitative results. All methods are evaluated under identical random seed protocols and fixed data budgets for fair comparison.

5.2 Experiment I: Zero-Shot Generalization & Effectiveness

Experimental Question: The experiments in this section evaluate the effectiveness of MAC in improving the world model's accuracy in simulating the domain dynamics, and whether training a transferable world model by MAC can achieve better or more robust performance when transferred to novel environments. To study these questions, we implemented experiments on MiniGrid shown in Figure 5.

5.2.1 Experimental Setup.

Baselines. We compare our proposed Model-driven Adversarial Curriculum(MAC) framework against two distinct baselines representing different training paradigms:

Target Direct Learning (Baseline) In this setting, the World Model is trained directly on the Target Tasks sequence using a random exploration policy. We maintain the same total data volume and continual learning mechanism (Fisher Replay Buffer) as the curriculum method to ensure a fair comparison. This baseline investigates whether learning from adaptively generated minitasks yields better generalization than attempting to learn directly from the complex target environments, where unguided random exploration often fails to cover diverse or critical state transitions efficiently.

[1]The repository is anonymized for double-blind review and will be made publicly available after publication.

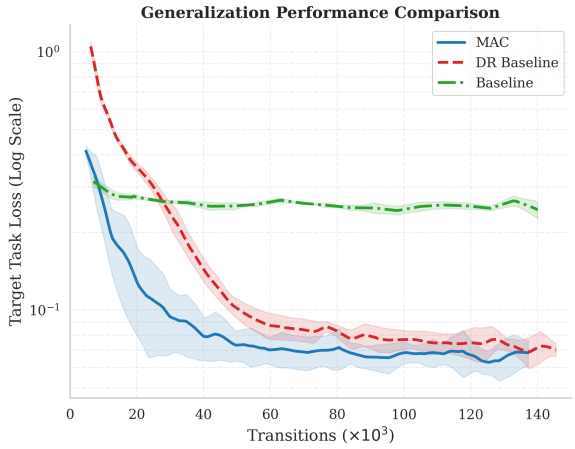

Figure 7: Zero-shot generalization performance (validation MSE on held-out Target Tasks) for MAC, Domain Randomization (DR), and direct training on Target Tasks (Baseline).

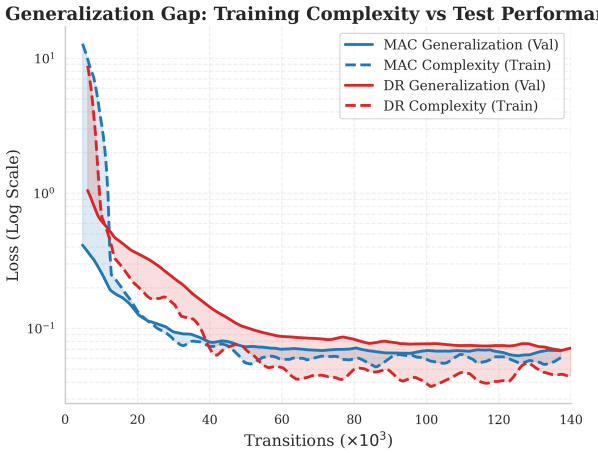

Figure 8: Training dynamics decomposition comparing MAC and Domain Randomization (DR).

The final performance is evaluated on the separate Uniform Exploration Dataset to verify the true learning effect.

Random Generator (Domain Randomization) Instead of an adversarial editor, we use a random agent that performs stochastic edits on the minitasks subject to the same budget constraint, $\rho = 0.5$. This baseline (Domain Randomization) evaluates the contribution of the curriculum itself, verifying that the performance gains stem from the adversarial targeting of weaknesses rather than mere data augmentation.

5.2.2 Results and Analysis. In this section, we evaluate the performance of our proposed Model-driven Adversarial Curriculum (MAC) against two baselines: (i) *Domain Randomization (DR)* (Random Generator) and (ii) a *Baseline* that trains the World Model directly on the Target Tasks via using a fixed, uniformly random exploration policy (Target Direct Learning). Our primary objective is to assess the ability of the learned World Model to *generalize zero-shot* to unseen, complex target tasks after being trained on curriculum-generated minitasks.

Generalization Performance. We measure generalization by the validation Mean Squared Error (MSE) on the held-out Target Task Set (Fig. 6), which is never used for training. Throughout, we use the same interaction-aware *weighted* prediction loss for both training and reporting validation curves, so that errors on sparse but crucial agent–object interactions are not dominated by static background pixels. Figure 7 compares the validation loss curves across training transitions. MAC converges substantially faster than both baselines, indicating that the adversarial editor consistently identifies *critical learning samples*—maps that lie near the frontier of the current World Model's predictive capability. DR can eventually reach a comparable validation loss, but requires substantially more transitions to do so, highlighting MAC's superior sample efficiency. This also suggests that, at the current MiniGrid difficulty, random generation can still traverse a large portion of the key dynamics given sufficient data.

In contrast, the Target Direct Learning Baseline is consistently the least sample-efficient under a fixed data budget: because it must collect training data directly in the complex target environments, a random exploration policy covers interaction-critical transitions very slowly, yielding poor learning progress when compared to MAC.

Analysis of Training Dynamics: Generalization Gap. To understand why MAC outperforms DR despite both producing "random-looking" maps, we analyze the training dynamics through the lens of the Generalization Gap (Fig. 8). In this panel, the dashed curves report the raining Complexity (loss on newly generated tasks), while the solid curves report the Generalization (validation loss on the fixed Target Task Set).

MAC (blue) exhibits a superior curriculum pattern. In the early phase (transitions < 20k), the Training Complexity stays above the validation curve, indicating that MAC actively generates tasks that are harder than the current target distribution to accelerate learning. Even later, as the World Model improves and becomes difficult to "adversarialize," MAC prevents the generator from degrading to trivial tasks. Instead, it maintains the generated-task difficulty on par with the target validation level. This alignment is visually confirmed by the minimal blue generalization gap (shaded area), showing that the generated challenges remain highly relevant to the target domain.

In stark contrast, DR (red) ceases to generate tasks harder than the target distribution as early as 15k transitions, where its complexity curve dives significantly below the validation curve. This premature drop demonstrates the ineffectiveness of random generation: it fails to track the model's growing capabilities, creating an expansive generalization gap (red shaded region). While MAC sustains relevant challenges, DR allows the model to overfit to simple tasks without mastering the actual target dynamics.

Key finding. Overall, MAC is not merely a "harder" training regime, but a more targeted one: by focusing on epistemic weaknesses rather than superficial variability, it distinguishes learnable

structure from uninformative noise and yields substantially improved zero-shot generalization. We note, however, that MiniGrid may still be relatively simple, so performance gaps can become less pronounced in the high-data regime where random generation can eventually cover many key interactions. In the future work, we will evaluate MAC on more complex domains to further stress-test curriculum-driven world model learning, including 3D and stochastic ones.

5.3 Experiment II: Ablation Study

Experimental Question: To evaluate the contribution of each core component in the Full MAC framework, we conduct a series of ablation experiments focusing on zero-shot generalization.

5.3.1 Ablation Setup. We compare the Full MAC method against two key variants to isolate the effects of history and diversity:

- **Full MAC (Ours):** The complete configuration where the generator policy $\pi_\phi(a_e \mid M, c)$ is conditioned on both the current configuration M and an epistemic feedback embedding $c = f_\omega(M, \mathcal{E})$, which encodes aggregated world model prediction errors. The generator is optimized using the full curriculum reward including adversarial error, validity, and diversity terms.
- **w/o History (Green):** A conditioning-level ablation where the epistemic feedback encoder f_ω is removed. The generator policy is conditioned solely on the current configuration M, without access to historical prediction error information. The reward formulation remains unchanged.
- **w/o Diversity (Orange):** An objective-level ablation where the diversity reward term $R_{\text{diversity}}$ is removed from the generator objective. The conditioning structure, including the epistemic feedback embedding, remains identical to the Full MAC configuration.

All variants are evaluated using the Zero-shot Validation Loss, ensuring a fair comparison under identical hyperparameter settings.

5.3.2 Results and Analysis. Figure 9 reports zero-shot generalization performance across ablations. The Full MAC configuration (blue) consistently achieves the lowest validation loss throughout training. Although the *w/o History* variant (green) performs comparably during early iterations, Full MAC begins to diverge slightly after iteration 15 and maintains a modest advantage until convergence. This suggests that epistemic feedback conditioning is not essential for bootstrapping early dynamics learning, but contributes to incremental refinement of the curriculum in later stages.

The *w/o History* variant eventually plateaus at a slightly higher validation loss. Without the epistemic feedback embedding $c = f_\omega(M, \mathcal{E})$, the generator does not explicitly condition on aggregated prediction error patterns. In the current MiniGrid setting, this results in a modest but consistent performance gap relative to the full configuration, indicating that history provides additional refinement benefits, though its contribution remains moderate at this level of domain complexity.

In contrast, the *w/o Diversity* variant (orange) performs strictly worse than the other configurations. Removing the diversity reward $R_{\text{diversity}}$ reduces structural coverage in the configuration space,

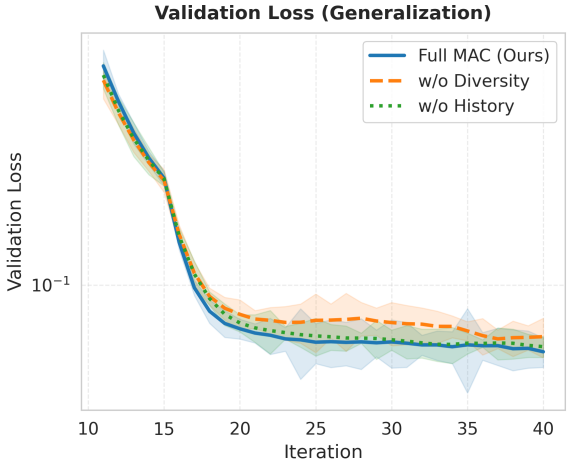

Figure 9: Ablation study on MiniGrid. We report held-out validation MSE (zero-shot generalization) for different MAC variants. Full MAC consistently achieves the lowest validation loss throughout training.

causing the generator to repeatedly exploit similar adversarial patterns. This limits exposure to heterogeneous transition regimes and leads to higher validation error and increased instability.

Overall, Full MAC achieves the most robust trade-off. Epistemic feedback conditioning enables targeted frontier tracking, while the diversity objective ensures broad structural exploration. The combination of these two mechanisms is necessary to avoid both undirected curriculum stagnation and structural over-specialization.

6 CONCLUSION

In this paper, we proposed Model-driven Adversarial Curriculum (MAC), a closed-loop framework that actively generates training environments to accelerate world model learning. By coupling procedural generation with an adversarial editor guided by epistemic feedback, MAC targets underrepresented and poorly predicted transitions and enables a single continually trained world model to accumulate reusable dynamics knowledge. We further introduced an Agent-Centric Attentive World Model (AC-AWM) that improves transfer by learning action-conditioned, agent-centric dynamics representations.

The initial empirical results on MiniGrid demonstrate that MAC improves predictive accuracy and robustness, and yields world models that generalize better to unseen target environments than direct training on target tasks and random environment generation. In particular, MAC achieves these gains with markedly higher sample efficiency, whereas random generation can require substantially more transitions to reach comparable performance.

A limitation of our current evaluation is that it focuses on a relatively simple grid-world domain, where random generation may eventually cover many key interactions in the high-data regime. In future work, we will broaden the scope of MAC beyond maze-like environments by (i) extending the environment editor and world model to richer, partially observable settings such as Crafter, and (ii) applying MAC to continuous-control robotics benchmarks such

as MetaWorld to assess scalability and generality. These directions will clarify how model-driven curricula can support generalized world models in more complex visual and robotic environments.

REFERENCES

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum Learning. In *Proceedings of the 26th International Conference on Machine Learning*.

[2] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. 2019. Exploration by Random Network Distillation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1lJJnR5Ym

[3] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. 2018. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. In *Advances in Neural Information Processing Systems*.

[4] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. 2020. Emergent Complexity and Zero-Shot Transfer via Unsupervised Environment Design. In *Advances in Neural Information Processing Systems*.

[5] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. 2017. Reverse Curriculum Generation for Reinforcement Learning. In *Conference on Robot Learning*.

[6] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2020. Dream to Control: Learning Behaviors by Latent Imagination. In *International Conference on Learning Representations (ICLR)*. Dreamer V1.

[7] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. 2019. Learning Latent Dynamics for Planning from Pixels. In *International Conference on Machine Learning (ICML)*. The "PlaNet" paper (Pre-Dreamer).

[8] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. 2021. Mastering Atari with Discrete World Models. In *International Conference on Learning Representations (ICLR)*. Dreamer V2.

[9] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. 2025. Mastering diverse control tasks through world models. *Nature* (2025). Dreamer V3.

[10] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. 2016. VIME: Variational Information Maximizing Exploration. In *Advances in Neural Information Processing Systems*.

[11] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. 2019. When to Trust Your Model: Model-Based Policy Optimization. In *Advances in Neural Information Processing Systems*.

[12] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[13] Yann LeCun. 2022. A Path Towards Autonomous Machine Intelligence. *arXiv preprint arXiv:2207.05601* (2022).

[14] Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. 2018. Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning. In *2018 IEEE International Conference on Robotics and Automation*.

[15] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. 2020. Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey. *Journal of Machine Learning Research* 21, 181 (2020), 1–50.

[16] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. 2017. Curiosity-Driven Exploration by Self-Supervised Prediction. In *Proceedings of the 34th International Conference on Machine Learning*.

[17] Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. 2020. Automatic curriculum learning for deep RL: A short survey. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*. 4819–4825.

[18] Sebastian Risi and Julian Togelius. 2020. Increasing generality in machine learning through procedural content generation. *Nature Machine Intelligence* 2, 8 (2020), 428–436.

[19] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2016. Prioritized Experience Replay. In *International Conference on Learning Representations*.

[20] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, and David Silver. 2020. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. *Nature* 588, 7839 (2020), 604–609.

[21] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 23–30.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*.

[23] Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. 2019. Paired Open-Ended Trailblazer (POET): Endlessly Generating Increasingly Complex and Diverse Learning Environments and Their Solutions. In *Genetic and Evolutionary Computation Conference*.

[24] Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. 2015. Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images. In *Advances in Neural Information Processing Systems*.

[25] Wojciech Zaremba and Ilya Sutskever. 2014. Learning to Execute. In *International Conference on Learning Representations (ICLR)*.

[26] Mingde Zhao, Zhen Liu, Sitao Luan, Shuyuan Zhang, Doina Precup, and Yoshua Bengio. 2021. A Consciousness-Inspired Planning Agent for Model-Based Reinforcement Learning. In *Advances in Neural Information Processing Systems*. arXiv:2106.02097.