RESPRIT: RESET-AWARE IMAGINATION FROM SELF-PREDICTIVE REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Model-free agents such as BBF achieve strong performance on Atari 100K, yet use their predictive models only to improve representations. Can imagination from a compact predictive model improve an already strong off-policy learner? We introduce Reset-aware Self-Predictive Representation Imagination Training (ReSPRIT), which adds reward and continuation prediction heads to BBF's existing latent prediction model to form a compact world model. Short imagined rollouts provide auxiliary actor and value learning alongside updates from replayed experience. Reset-aware scheduling limits the influence of unreliable imagined targets after parameter resets, while reduced imagination-loss weights balance learning from real and imagined experience. On Atari 100K, ReSPRIT improves the human-normalized interquartile mean (IQM) score from 1.025 for the matched baseline without imagination to 1.177, the highest reported canonical IQM among the compared agents. It also achieves a higher game-level IQM than EfficientZero V2, while requiring no lookahead search at inference. In our Asterix comparison, ReSPRIT completes training and evaluation in roughly one-third the time of EfficientZero V2. Ablation studies support the benefits of imagination and the importance of reset-aware scheduling and loss weighting.

1 INTRODUCTION

Data-efficient reinforcement learning (RL) asks how much an agent can learn from a tightly limited budget of environment interaction. On discrete-control benchmarks, this question is commonly posed on Atari 100K (Łukasz Kaiser et al., 2020): a suite of 26 Atari games in which the agent may collect only 100K environment steps, or about two hours of human play. The protocol was introduced with the model-based agent SimPLe (Łukasz Kaiser et al., 2020) and has since become the standard testbed for sample-efficient discrete RL.

One line of work stays model-free and improves an off-policy value learner. Data-efficient Rainbow (DER) (van Hasselt et al., 2019) showed that a carefully configured Rainbow agent (Hessel et al., 2018) is already a strong baseline under the 100K-step budget. Subsequent advances include image augmentation in DrQ (Yarats et al., 2021), contrastive representation learning in CURL (Laskin et al., 2020), prediction of future latent representations in SPR (Schwarzer et al., 2021), and parameter resets that enable higher replay ratios in SR-SPR (D'Oro et al., 2023). Building on this line of work, Bigger, Better, Faster (BBF) (Schwarzer et al., 2023) became the first model-free agent to exceed the human reference on Atari 100K's human-normalized interquartile mean (IQM), an aggregate measure across games and runs. SAC-BBF (Zhang et al., 2024) subsequently introduced an explicit policy while retaining replay-based critic learning and SPR.

Another approach uses learned world models to generate additional training trajectories. Dreamer, DreamerV2, and DreamerV3 learn behavior through latent imagination (Hafner et al., 2020; 2021; 2025). IRIS uses autoregressive token prediction (Micheli et al., 2023), DIAMOND uses diffusion-based visual prediction (Alonso et al., 2024), and TWISTER combines a Transformer state-space model with contrastive predictive coding (Burchi & Timofte, 2025). Recent agents further refine token-based world models for Atari 100K, including Simulus, EASimulus, and ITC (Cohen et al., 2026; Peng et al., 2026; Kim et al., 2026). Learned models can also support lookahead search: MuZero (Schrittwieser et al., 2020) and its sample-efficient successors, EfficientZero (EZ) (Ye et al., 2021) and EfficientZero V2 (EZ-V2) (Wang et al., 2024), use tree search to select actions.

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

These advances suggest combining the strengths of replay-based learning and imagination, but the benefit of doing so is not obvious. DreamerV3 baselines in recent Atari 100K comparisons achieve IQMs around 0.5, well below BBF's reported 1.045 (Cohen et al., 2026; Peng et al., 2026; Schwarzer et al., 2023). Recent agents such as EASimulus and ITC exceed an IQM of 1.0, but build on advances in token-based world modeling beyond DreamerV3 (Peng et al., 2026; Kim et al., 2026). These results leave open whether imagination from a much simpler model can improve an agent that already learns effectively from real experience.

The SPR-based agents discussed above offer a concrete starting point for investigating this question. SPR, SR-SPR, BBF, and SAC-BBF already train an action-conditioned model to predict latent representations several steps into the future. This model provides the dynamics component of a compact world model, although it lacks reward and continuation predictions. Its rollouts supervise representations rather than supply imagined returns for behavior learning. Building on this predictive structure, we ask: *Can a strong off-policy learner benefit from imagined experience generated by the compact model it already has?*

We introduce Reset-aware Self-Predictive Representation Imagination Training (ReSPRIT), which extends the SPR-based BBF/SAC-BBF framework with imagination-based learning. ReSPRIT adds reward and continuation prediction heads to the existing latent dynamics, yielding a compact, deterministic world model. Short rollouts under the current policy then provide auxiliary actor and action-value learning targets alongside the agent's replay-based updates. This construction reuses both the latent transition model and the action-value function, without requiring a pixel decoder, a separate state-value network, or lookahead search at inference.

For imagination to help, its learning signals must complement the updates from real experience. Imperfect imagined returns can disrupt this balance, particularly after the parameter resets inherited from BBF alter the latent dynamics and reinitialize the policy and value heads. ReSPRIT therefore retains the reward and continuation readouts and temporarily suppresses imagination losses after initialization and each reset. It then gradually increases their weights as replay-based learning continues, keeping their final values small to limit the influence of imperfect imagined targets.

We make two main contributions:

1. We show that a strong off-policy learner benefits from imagined actor and value learning using a compact SPR-based world model. ReSPRIT reuses the learner's latent dynamics and action-value function, with ablations supporting reset-aware scheduling and reduced imagination-loss weights.

2. ReSPRIT achieves a canonical Atari 100K IQM of 1.177, exceeding the matched baseline (1.025) and the reported canonical IQMs of the compared agents. It also surpasses EZ-V2 in game-level IQM. In our same-hardware Asterix comparison, training and evaluation take approximately 18% longer than the matched baseline and one-third of EZ-V2's time, with no lookahead search at inference.

2 PRELIMINARIES

We consider episodic RL in a Markov decision process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma)$, where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, $P(s' \mid s, a)$ is the probability of transitioning to state s' after taking action a in state s, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, ρ_0 is the initial-state distribution, and $\gamma \in (0, 1)$ is the discount factor. A stochastic policy $\pi(a \mid s)$ specifies a probability distribution over actions in each state. The objective is to learn a policy that maximizes the expected discounted return over an episode: $J(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{\tau-1} \gamma^t r(s_t, a_t)\right]$, where τ is the episode length, $s_0 \sim \rho_0$, $a_t \sim \pi(\cdot \mid s_t)$, and $s_{t+1} \sim P(\cdot \mid s_t, a_t)$. Below, we review the components of BBF and SAC-BBF on which ReSPRIT builds.

We use θ collectively for the trainable network parameters and the superscript targ for moving-average target networks. When Q_θ or π_θ takes a state s as input, it includes the encoder; when it takes a latent representation z, it denotes the corresponding prediction head.

Replay-based value learning. A core component of BBF is n-step Q-learning with experience replay. For a sampled trajectory segment with update horizon n, the underlying scalar TD target is

$$y_t = r_t^{(n)} + \gamma^n \left[\prod_{j=1}^{n}(1 - d_{t+j})\right] Q_\theta^{\text{targ}}(s_{t+n}, a^*), \qquad a^* = \arg\max_{a \in \mathcal{A}} Q_\theta(s_{t+n}, a), \tag{1}$$

where $r_{t+1} = r(s_t, a_t)$ is the immediate reward, $r_t^{(n)} = \sum_{k=0}^{n-1} \gamma^k \left[\prod_{j=1}^{k}(1 - d_{t+j})\right] r_{t+k+1}$ is the discounted reward accumulated up to the update horizon or episode termination, and $d_{t+j} \in \{0, 1\}$ indicates whether the transition from s_{t+j-1} under action a_{t+j-1} terminates the episode. The online network selects the bootstrap action, while the target network evaluates it. BBF implements this backup using a categorical return distribution supported on 51 atoms and minimizes the cross-entropy between the projected target distribution and the predicted distribution.

During training, BBF periodically resets network parameters to maintain plasticity, enabling replay ratios of up to 8. The replay ratio is the number of gradient updates per environment step. BBF also adjusts n and γ in Eq. 1 following each reset: over the first 10K gradient steps, n decreases exponentially from 10 to 3, while γ increases from 0.97 to 0.997.

Self-predictive representations (SPR). BBF builds on SR-SPR, which uses self-predictive representations (SPR) to train the agent to predict its own latent state representations multiple steps into the future.

Given a replay sequence starting at s_t, the SPR module predicts H future latent representations by iteratively applying a transition model conditioned on the recorded actions:

$$\begin{aligned} \hat{z}_t &= z_t = e_\theta(s_t), \\ \hat{z}_{t+k} &= f_\theta(\hat{z}_{t+k-1}, a_{t+k-1}), \qquad 1 \le k \le H, \end{aligned} \tag{2}$$

where e_θ is an encoder implemented as a widened IMPALA-style CNN, and f_θ is a convolutional transition model. To condition each prediction on the action, the one-hot encoding of a_{t+k-1} is broadcast across the spatial grid of $\hat{z}_{t+k-1}$ and concatenated along the channel dimension.

The target representations $\bar{z}_{t+1:t+H}$ are obtained by applying the target encoder e_θ^{targ} to the observed future states $s_{t+1:t+H}$. The SPR objective encourages high cosine similarity between the projected predictions $q_\theta(g_\theta(\hat{z}_{t+k}))$ and the target representations $g_\theta^{\text{targ}}(\bar{z}_{t+k})$ over valid prediction steps within the same trajectory, where g_θ and q_θ are projection and prediction heads implemented as simple MLPs, and g_θ^{targ} is the target projection head.

Policy learning in SAC-BBF. BBF selects actions with an ϵ-greedy policy derived from its Q-values. SAC-BBF (Zhang et al., 2024) adds an explicit policy head π_θ. Its implementation uses an n-step Sarsa-style backup from replay: the bootstrap action in Eq. 1 is sampled from $\pi_\theta(\cdot \mid s_{t+n})$ instead of being selected greedily, while the accumulated rewards still come from replay. For policy learning, a fresh action $a' \sim \pi_\theta(\cdot \mid s_t)$ is sampled at each replay state s_t. The actor minimizes

$$\begin{aligned} A_\theta(s_t, a') &= Q_\theta(s_t, a') - \sum_{a \in \mathcal{A}} \pi_\theta(a \mid s_t) Q_\theta(s_t, a), \\ \mathcal{L}_\pi^{\text{replay}}(s_t) &= -\,\text{sg}(A_\theta(s_t, a')) \log \pi_\theta(a' \mid s_t) - \eta \mathcal{H}(\pi_\theta(\cdot \mid s_t)), \end{aligned} \tag{3}$$

where sg denotes stop-gradient and $\mathcal{H}$ is policy entropy. The action-dependent learning signal uses the freshly sampled a', rather than the recorded replay action a_t, and the policy-weighted Q-value baseline reduces gradient variance. The coefficient η decays linearly from 10^{-2} to zero over 80K environment steps.

3 Main techniques

ReSPRIT turns SPR's latent dynamics into a compact world model by adding reward and continuation prediction heads (Fig. 1). Short policy-generated rollouts then provide additional training signals for the existing actor and Q-network. State-value estimates are derived from these networks, avoiding the separate value network used in DreamerV3. We describe the prediction losses, imagination objectives, and schedules that coordinate this learning with replay and parameter resets.

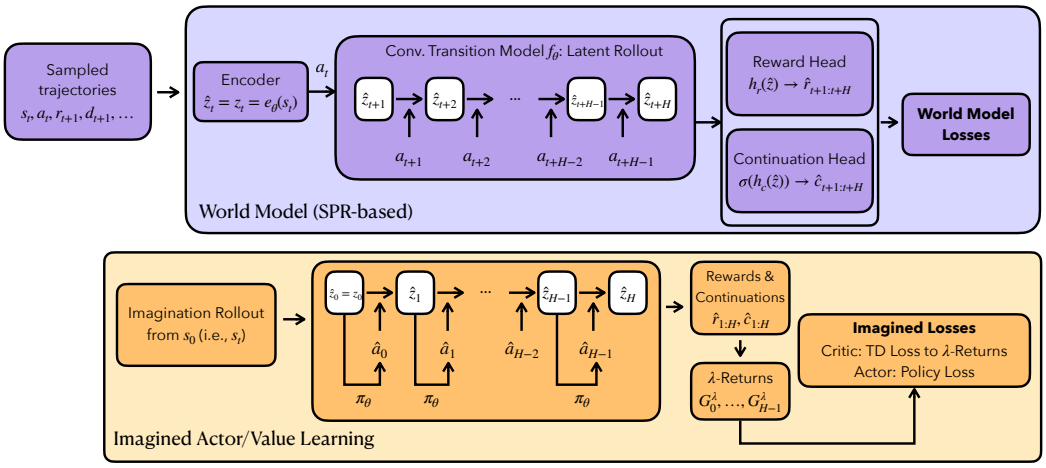

Figure 1: World-model training and imagination learning in ReSPRIT. Top: SPR dynamics follow actions recorded in replay, and the added reward and continuation heads learn from the corresponding observed outcomes. Bottom: a separate rollout begins at the same replay state s_t and follows newly sampled policy actions. Predicted rewards, continuation probabilities, and bootstrap values form λ-returns for auxiliary actor and action-value learning. The bottom panel uses relative time indices, with step 0 corresponding to replay time t.

3.1 A COMPACT WORLD MODEL

We retain the base agent's encoder e_θ and SPR transition model f_θ, and add reward and continuation predictors. We use arrival indices for transition outcomes: r_{t+k} and d_{t+k} are the reward and training terminal flag produced by taking action a_{t+k-1} in state s_{t+k-1}. Following BBF, rewards used for training are clipped to $[-1, 1]$.

We augment these latent dynamics with two scalar MLP heads:

$$\hat{r}(z) = h_r(z), \qquad \hat{c}(z) = \sigma(h_c(z)), \tag{4}$$

where σ denotes the sigmoid function and the spatial latent is flattened before entering either head. Applied to $\hat{z}_{t+k}$ in Eq. 2, these heads predict the reward r_{t+k} and the probability that the transition arriving at $t + k$ is nonterminal. We train the reward head by squared-error regression. In addition to predicted latents, the reward head is trained on representations of observed future states, $\bar{z}_{t+k} = e_\theta^{\text{targ}}(s_{t+k})$, produced by the target encoder.

Training these two heads requires care at trajectory boundaries. We want to learn the reward and termination of the terminal transition itself, while excluding predictions beyond that boundary. More precisely, suppose the first terminal transition occurs when taking action a_{t+k-1}, so that $d_{t+k} = 1$. We still train the reward and continuation heads on $\hat{z}_{t+k}$ to predict r_{t+k} and $1 - d_{t+k} = 0$, respectively. However, we exclude the corresponding target-encoded representation $\bar{z}_{t+k}$ from reward supervision: the stored state s_{t+k} is not guaranteed to be the terminal observation and may instead belong to a reset trajectory. To distinguish these cases, define

$$u_0 = 1, \qquad u_k = \prod_{j=1}^{k} (1 - d_{t+j}), \qquad m_k = u_{k-1}, \qquad 1 \le k \le H. \tag{5}$$

Here u_k indicates that no terminal transition has occurred through step k. At the first terminal transition, $m_k = 1$ and $u_k = 0$; for all subsequent steps, both masks are zero. We therefore use m_k for supervision on predicted latents $\hat{z}_{t+k}$, retaining the terminal transition while excluding all later predictions. For reward supervision on target-encoded representations $\bar{z}_{t+k}$, we use $m_k u_k$, which additionally excludes the terminal arrival.

Let $\text{MM}[x, m]$ denote the mean of loss values x selected by mask m, over prediction steps $1 \le k \le H$ and sampled sequences, with value zero when no entries are retained. Omitting replay importance

weights for clarity, the readout losses are

$$\mathcal{L}_r = \mathrm{MM}\Big[\big(\hat{r}(\mathrm{sg}(\hat{z}_{t+k})) - r_{t+k}\big)^2, m_k\Big] + \mathrm{MM}\Big[\big(\hat{r}(\mathrm{sg}(\bar{z}_{t+k})) - r_{t+k}\big)^2, m_k u_k\Big], \tag{6}$$

$$\mathcal{L}_c = \mathrm{MM}[\mathrm{BCE}(\hat{c}(\mathrm{sg}(\hat{z}_{t+k})), u_k), m_k], \tag{7}$$

where $\mathrm{BCE}(p, y)$ is binary cross-entropy between a predicted probability p and a binary target y. We omit continuation supervision on $\bar{z}_{t+k}$ because excluding terminal arrivals leaves only nonterminal examples. This additional loss would encourage higher continuation predictions without providing terminal examples.

In training, each loss value receives the replay weight of its starting state before the masked mean is taken; the masks and their normalization are unchanged (Section 3.4). The stop-gradient operations in Eqs. 6 and 7 restrict these losses to the reward and continuation heads. The encoder and transition model continue to learn through the base agent's objectives. Imagined actor and value updates also leave the world model unchanged. The resulting model retains SPR's compact, deterministic dynamics and requires neither a pixel decoder nor a stochastic recurrent state-space model.

3.2 Imagined actor and value learning

We use the world model for actor-critic learning in imagination, following DreamerV3. Rather than training a separate state-value network, we average the existing target Q-values under the current policy:

$$V_{t+k} = \sum_{a \in \mathcal{A}} \mathrm{sg}(\pi_\theta(a \mid \hat{z}_{t+k}))\, Q_\theta^{\mathrm{targ}}(\hat{z}_{t+k}, a). \tag{8}$$

For each sampled segment, we initialize an H-step imagined rollout from its starting state s_t. In this subsection, $\hat{z}_{t+k}$ denotes the latents of this separate rollout, which follows actions $\hat{a}_{t+k-1}$ sampled from π_θ instead of the recorded actions used in Eq. 2:

$$\begin{aligned} \hat{z}_t &= \mathrm{sg}(e_\theta(s_t)), \\ \hat{a}_{t+k-1} &\sim \pi_\theta(\cdot \mid \hat{z}_{t+k-1}), \\ \hat{z}_{t+k} &= \mathrm{sg}(f_\theta(\hat{z}_{t+k-1}, \hat{a}_{t+k-1})), \end{aligned} \tag{9}$$

where the action-sampling and transition equations apply for $1 \le k \le H$.

The resulting latents provide reward and continuation predictions $\hat{r}_{t+k} = \hat{r}(\hat{z}_{t+k})$ and $\hat{c}_{t+k} = \hat{c}(\hat{z}_{t+k})$ for $1 \le k \le H$, and value estimates V_{t+k} for $0 \le k \le H$. Each rollout starts from an active state, and we set $\hat{c}_t = 1$.

Following DreamerV3, we construct bootstrapped λ-returns, with $\lambda \in [0, 1]$, for actor-critic learning:

$$\begin{aligned} G^\lambda_{t+H} &= V_{t+H}, \\ G^\lambda_{t+k} &= \hat{r}_{t+k+1} + \gamma \hat{c}_{t+k+1} \left[(1-\lambda) V_{t+k+1} + \lambda G^\lambda_{t+k+1}\right], \end{aligned} \tag{10}$$

for $k = H-1, \ldots, 0$. Here γ is the same discount used in replay-based TD learning. Both actor and critic losses apply at depths $0 \le k < H$, while the endpoint supplies the bootstrap value. Thus, the return at depth k uses up to $H - k$ predicted rewards together with value estimates.

We use a shared horizon H for world-model training and imagination. This aligns the depth of imagined rollouts with the multi-step supervision provided by SPR, avoiding additional extrapolation to prediction depths beyond those used to train the latent dynamics.

3.2.1 Imagined actor loss

Following DreamerV3, we normalize actor advantages using a running estimate of the return scale. Let $\mathcal{G}$ contain the λ-returns at all H training positions ($k = 0, \ldots, H-1$) across the current minibatch. We maintain an exponential moving average of their 5th-to-95th percentile range:

$$S = \mathrm{EMA}(\mathrm{Per}(\mathcal{G}, 95) - \mathrm{Per}(\mathcal{G}, 5),\ 0.99). \tag{11}$$

Here $\mathrm{Per}(\mathcal{G}, p)$ is the pth percentile, and the moving average uses decay 0.99. The advantage $G^\lambda_{t+k} - V_{t+k}$ is divided by $\max(1, S)$, which prevents small return ranges from amplifying the learning signal.

We use a REINFORCE surrogate to improve the expected discounted return from s_t. Each actor-loss term is weighted by discounted survival:

$$\xi_0 = 1, \qquad \xi_k = \prod_{j=1}^{k} (\gamma \hat{c}_{t+j}) = \gamma^k \prod_{j=1}^{k} \hat{c}_{t+j}, \qquad 1 \le k < H. \tag{12}$$

The continuation product accounts for predicted survival to depth k, while γ^k places each term on the starting state's discount scale. The bootstrapped λ-returns provide approximate advantages for this surrogate.

The actor loss for one imagined trajectory is

$$\mathcal{L}_\pi^{\text{imag}} = -\frac{1}{H} \sum_{k=0}^{H-1} \text{sg}(\xi_k) \left[\log \pi_\theta(\hat{a}_{t+k} \mid \hat{z}_{t+k}) \, \text{sg}\left(\frac{G_{t+k}^\lambda - V_{t+k}}{\max(1, S)} \right) + \eta \, \mathcal{H}(\pi_\theta(\cdot \mid \hat{z}_{t+k})) \right], \tag{13}$$

where η is the same entropy regularization coefficient used in ReSPRIT's replay-based actor learning.

3.2.2 Imagined action-value learning

The same detached λ-returns supervise the online Q-network at imagined state-action pairs. Let $p_\theta(\cdot \mid z, a)$ denote its categorical return distribution, whose expectation is $Q_\theta(z, a)$. We project each scalar return onto the existing 51-atom C51 support and minimize the cross-entropy loss:

$$\mathcal{L}_Q^{\text{imag}} = \frac{1}{H} \sum_{k=0}^{H-1} \text{sg}(\xi_k) \, \text{CE}\big(\text{Proj}_{\text{C51}}\big(\text{sg}(G_{t+k}^\lambda)\big), p_\theta(\cdot \mid \hat{z}_{t+k}, \hat{a}_{t+k})\big), \tag{14}$$

where Proj_{C51} maps a scalar return to a categorical target on the Q-network's categorical support, and $\text{CE}(p, q) = -\sum_i p_i \log q_i$. Both losses are averaged over sampled trajectories. The imagined latents and return targets are detached, so these objectives update the policy and Q-network without modifying the world model.

3.3 Reset-aware imagination scheduling

At replay ratio 2, the inherited BBF plasticity mechanism resets the network at approximately 20K, 40K, and 60K environment steps. At each reset, the encoder e_θ and convolutional transition model f_θ undergo shrink-and-perturb updates that combine their learned parameters with freshly initialized parameters, assigning 50% weight to each. The policy and Q-network projections and output heads are fully reinitialized.

We deliberately exempt the reward and continuation heads from both reinitialization and shrink-and-perturb, retaining their parameters unchanged across resets. These heads supply the reward and continuation signals used for imagination learning; preserving them avoids discarding or perturbing the predictive information they have already acquired.

These resets change the latent representations on which the retained readouts operate, as well as the policy generating imagined actions and the Q-values used to bootstrap their returns. Consequently, imagination-based updates may be unreliable immediately after a reset. We therefore temporarily suppress their contribution and restore it gradually as learning proceeds.

Let g count gradient updates since initialization or the most recent parameter reset. We multiply both $\mathcal{L}_\pi^{\text{imag}}$ and $\mathcal{L}_Q^{\text{imag}}$ by

$$a(g) = \text{clip}\left(\frac{g - W}{W}, 0, 1 \right), \tag{15}$$

where $W > 0$ specifies the duration of the initial suppression interval. The imagination losses contribute nothing for $g \le W$, increase linearly in weight for $W < g < 2W$, and reach their full weight at $g = 2W$.

Replay-based TD and actor learning, SPR, and supervised reward and continuation learning remain active throughout this schedule. The zero-weight interval therefore allows the latent dynamics,

readouts, and value estimates to readapt before imagined targets contribute to policy and Q-network updates.

The running percentile statistics used for return normalization are preserved across resets and held fixed while $a(g) = 0$. Their updates resume when imagination learning becomes active. This retains previously accumulated normalization information without incorporating return statistics generated during the suppression interval.

3.4 Prioritized replay and shared schedules

ReSPRIT applies prioritized experience replay (PER) consistently across its learning objectives. Each prioritized sampling operation supplies two minibatches for successive gradient updates. For a sampled segment starting at s_t, the replay-based TD and actor losses are evaluated at s_t, SPR and world-model learning use the sampled segment starting at s_t, and imagination learning uses a policy-generated rollout initialized from s_t. All loss terms derived from this segment therefore receive the same importance-sampling weight associated with its starting state s_t.

Let $\rho(s_t)$ denote the stored sampling priority associated with s_t, and let $\mathcal{B}$ contain the starting states in the two jointly sampled minibatches. We define

$$w(s_t) = \frac{(\rho(s_t) + \epsilon)^{-\beta}}{\max_{s \in \mathcal{B}} (\rho(s) + \epsilon)^{-\beta}}, \qquad \beta = \frac{1}{2}, \tag{16}$$

where $\epsilon > 0$ ensures numerical stability. The inverse-priority weight partially compensates for nonuniform replay sampling, while normalization bounds the weights by one. Each sampled starting state's priority is updated using the square root of its unweighted C51 cross-entropy loss, with a small numerical stabilizer. The weight $w(s_t)$ multiplies the replay TD and actor losses and every SPR, reward, continuation, and imagination loss term derived from that starting state, before reduction over the minibatch.

Replay-based TD learning and imagination share the same discount schedule, while the replay and imagined actor objectives share the same entropy regularization schedule. We retain BBF's exponential decrease of $1 - \gamma$ and use an exponential decay for η over the same interval, replacing SAC-BBF's linear entropy schedule. Using the same update counter g as above, let T denote the schedule duration and $\chi(g)$ its normalized progress:

$$\begin{aligned} \chi(g) &= \mathrm{clip}\Big(\frac{g}{T}, 0, 1\Big), \\ \gamma(g) &= 1 - 0.03 \times 10^{-\chi(g)}, \\ \eta(g) &= 10^{-2} \times 10^{-\chi(g)}. \end{aligned} \tag{17}$$

Over T gradient updates, γ increases from 0.97 to 0.997, while η decreases exponentially from 10^{-2} to 10^{-3}. Both remain at their final values thereafter and restart following a parameter reset.

4 Experiments

We evaluate ReSPRIT on Atari 100K through comparisons with competitive agents and ablations of its main components. We identified several bugs in the official BBF implementation, also inherited by SAC-BBF. After correcting them, we were unable to reproduce the reported SAC-BBF results and therefore configured a matched baseline, as detailed in Appendix A. We present comparisons with other agents below; the matched-baseline comparison, ablations, and runtime measurements appear in Appendices B and E.

4.1 Comparison with competitive baselines

The comparisons cover model-free agents, world-model agents that learn through imagination, and agents that use lookahead search. Table 1 includes DER and SR-SPR as model-free references; DreamerV3, TWISTER, and DIAMOND as representative world-model agents; and Simulus, EASimulus, and ITC as recent advances in token-based world modeling. Comparisons with BBF and SAC-BBF appear alongside our matched baseline in Table 3.

Table 1: Atari 100K performance. DER and SR-SPR aggregates follow Schwarzer et al. (2023), Table A.1; other baseline values follow their respective papers. ReSPRIT values are computed from Table 5. IQM and optimality gap use canonical aggregation (Appendix A.4). Bold marks the best available result in each row; dashes indicate unavailable metrics.

Metric	DER	SR-SPR	DIAMOND	DreamerV3	TWISTER	Simulus	EASimulus	ITC	ReSPRIT
IQM ↑	0.183	0.631	0.641	–	–	0.990	1.004	1.092	**1.177**
Mean ↑	0.35	1.272	1.459	1.25	1.62	1.645	1.818	1.616	**2.724**
Median ↑	0.189	0.685	0.373	0.49	0.77	0.982	0.773	0.978	**0.990**
Optimality gap ↓	0.698	0.433	–	–	–	0.412	0.394	0.376	**0.333**
Games > human ↑	2	9	11	7	12	**13**	12	**13**	**13**

Table 2: Performance point estimates for EZ, EZ-V2, and ReSPRIT. EZ's published aggregates and per-game means follow Schwarzer et al. (2023), Table A.1; EZ-V2's follow Wang et al. (2024). Game-level IQM and optimality gap aggregate per-game means; canonical metrics aggregate per-run scores. EZ-V2's canonical IQM and optimality gap are unavailable for this analysis. Bold marks the best available point estimate in each column.

Method	IQM ↑		Mean ↑	Median ↑	Optimality gap ↓		Games > human ↑
	Canonical	Game-level			Canonical	Game-level	
EZ	1.02	1.051	1.945	1.116	0.371	0.367	14
EZ-V2	-	1.305	2.428	**1.286**	-	0.339	**15**
ReSPRIT	**1.177**	**1.352**	**2.724**	0.990	**0.333**	**0.291**	13

Finally, Table 2 compares ReSPRIT with EZ and EZ-V2, which use learned models for lookahead search. EZ-V2's per-run Atari 100K scores were unavailable for this comparison, consistent with the maintainer's response to a request for these scores. We therefore report its game-level IQM and optimality gap, computed from the published per-game means, but cannot compute the corresponding canonical metrics. The official repository's README also notes that the released implementation does not match the paper's results on some Atari games.

Among the available point estimates in Table 1, ReSPRIT has the highest IQM, mean, and median and the lowest optimality gap, while tying for the largest number of games above the human reference. Table 2 shows complementary strengths: EZ and EZ-V2 have higher medians and exceed the human reference on more games, whereas ReSPRIT has a higher mean. ReSPRIT's IQM is higher and its optimality gap lower than EZ's under both aggregations. Relative to EZ-V2, ReSPRIT has a higher game-level IQM (1.352 versus 1.305) and a lower game-level optimality gap (0.291 versus 0.339).

The matched-baseline comparison directly tests the benefit of adding imagination. ReSPRIT raises canonical IQM from 1.025 to 1.177 with the same approximately 240K gradient updates (Table 3). Thus, the gain comes from augmenting these updates with imagined learning, while retaining the baseline's replay schedule. Appendix B examines this improvement and the roles of reset-aware scheduling and imagination-loss weights.

5 Related Work

Replay-based learning and representations. DQN and Rainbow established pixel-based value learning, which DER adapted to limited interaction (Mnih et al., 2015; Hessel et al., 2018; van Hasselt et al., 2019). DrQ and CURL improve representations through augmentation and contrastive learning, respectively (Yarats et al., 2021; Laskin et al., 2020). SPR predicts future action-conditioned latents; SR-SPR adds parameter resets for higher replay ratios (Schwarzer et al., 2021; D'Oro et al., 2023). BBF scales networks and refines training schedules, while SAC-BBF adds an explicit policy to replay-based critic learning and SPR (Schwarzer et al., 2023; Zhang et al., 2024).

World models for imagination. SimPLe learns policies with video-prediction models on Atari 100K (Łukasz Kaiser et al., 2020). Dreamer learns behavior through imagined trajectories in recurrent stochastic latent models, with DreamerV3 targeting robustness across tasks (Hafner et al., 2020; 2021; 2025). DIAMOND generates frames through conditional diffusion and predicts rewards and termination with a separate CNN-LSTM (Alonso et al., 2024). TWISTER combines Transformer state-space modeling with action-conditioned contrastive predictive coding (Burchi & Timofte, 2025).

Token-based world models. IRIS pairs a discrete image autoencoder with an autoregressive Transformer (Micheli et al., 2023). Simulus combines flexible tokenization, uncertainty-driven exploration, prioritized world-model replay, and classification-based reward and return prediction (Cohen et al., 2026). EASimulus adds event prediction and segmentation to capture observational changes (Peng et al., 2026). ITC modifies Simulus decoding for Atari 100K, using optimal transport to coordinate copying previous-frame tokens versus generating new ones, without changing the backbone or training loss (Kim et al., 2026).

Lookahead search with learned dynamics. MuZero learns latent dynamics that predict rewards, policies, and values for Monte Carlo tree search (Schrittwieser et al., 2020). EZ improves data efficiency through self-supervised consistency, value-prefix prediction, and off-policy value-target correction (Ye et al., 2021). EZ-V2 extends this approach to discrete and continuous control (Wang et al., 2024).

6 Conclusion

We introduce ReSPRIT, which combines off-policy learning from replay with actor and value learning on imagined rollouts. By adding reward and continuation heads, ReSPRIT turns BBF's SPR module into a compact world model and improves performance over the matched baseline on Atari 100K. Our ablations highlight the importance of coordinating imagination with parameter resets and balancing its losses against replay-based learning.

A promising direction is to replace the SPR-based world model with a more expressive recurrent state-space model (RSSM), such as that used in DreamerV3. By summarizing interaction history in recurrent latent states, such a model could provide richer context for imagined predictions. Storing these states in replay and refreshing them during training could help retain this context when sampling trajectory segments. The official R2-Dreamer implementation (Morihira et al., 2026) provides a concise replay buffer with this functionality, offering a practical starting point for exploring the extension.

AI use statement

We used generative AI tools to improve the writing of this paper and to help locate recent state-of-the-art results on the Atari 100K benchmark. We also used these tools as coding assistants, including to review index alignment in our implementation of the λ-return in Eq. 10 and to help us better understand the BBF codebase. We did not use generative AI tools to formulate the research ideas, design the integration framework, or determine the ablation experiments. We reviewed all AI-assisted work and, where applicable, used mathematical derivations to verify that the implementation logic was consistent with our intended formulation. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

Reproducibility statement

To facilitate reproducibility, we report the raw scores from all 10 seeds in Table 5. The supplementary material includes the ReSPRIT code and concise installation and execution instructions for reproducing these results. Exact numerical agreement is not guaranteed, even with identical random seeds, because some GPU operations used in CNN training can be nondeterministic. The supplementary material also provides the raw scores from our reproduction of SAC-BBF (RR 4), the baseline described in Appendix A.2, and the ablation experiments summarized in Table 4. Ap-

pendix A details the modules added to ReSPRIT, the configuration changes relative to the publicly available BBF and SAC-BBF implementations, and the loss weights.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f514cec81cb148559cf475e7426eed5e-Paper.pdf.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=NadTwTODgC.

Maxime Burchi and Radu Timofte. Learning transformer-based world models with contrastive predictive coding. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YK9G4Htdew.

Lior Cohen, Kaixin Wang, Bingyi Kang, Uri Gadot, and Shie Mannor. Simulus: Combining improvements in sample-efficient world model agents, 2026. URL https://arxiv.org/abs/2502.11537.

Pierluca D'Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G Bellemare, and Aaron Courville. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OpC-9aBBVJe.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pp. 1–7, 2025.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Youngin Kim, Ray Sun, Inho Kim, Bumsoo Park, and Hyun Oh Song. Identifiable token correspondence for world models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=6vKlayY1tV.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: contrastive unsupervised representations for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Naoki Morihira, Amal Nahar, Kartik Bharadwaj, Yasuhiro Kato, Akinobu Hayashi, and Tatsuya Harada. R2-dreamer: Redundancy-reduced world models without decoders or augmentation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Je2QqXrcQq.

Zhao-Han Peng, Shaohui Li, Zhi Li, Shulan Ruan, Yu LIU, and You He. From observations to events: Event-aware world models for reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=OWkkFaq1IZ.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uCQfPZwRaUu.

Max Schwarzer, Johan Samir Obando Ceron, Aaron Courville, Marc G Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level atari with human-level efficiency. In *International Conference on Machine Learning*, pp. 30365–30380. PMLR, 2023.

Hado van Hasselt, Matteo Hessel, and John Aslanides. When to use parametric models in reinforcement learning? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1b742ae215adf18b75449c6e272fd92d-Paper.pdf.

Shengjie Wang, Shaohuai Liu, Weirui Ye, Jiacheng You, and Yang Gao. EfficientZero v2: Mastering discrete and continuous control with limited data. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 51041–51062. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wang24at.html.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=GY6-6sTvGaf.

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 25476–25488. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/d5eca8dc3820cad9fe56a3bafda65ca1-Paper.pdf.

Le Zhang, Yong Gu, Xin Zhao, Yanshuo Zhang, Shu Zhao, Yifei Jin, and Xinxin Wu. Generalizing soft actor-critic algorithms to discrete action spaces. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pp. 34–49. Springer, 2024.

Łukasz Kaiser, Mohammad Babaeizadeh, Piotr Miłos, Błażej Osiński, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for atari. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1xCPJHtDB.

A BASELINE AND RESPRIT AGENT CONFIGURATIONS AND EVALUATION METRICS

A.1 BUGS IDENTIFIED IN THE OFFICIAL BBF IMPLEMENTATION

We identified three bugs in the official BBF implementation, also inherited by SAC-BBF. Two concern indexing and alignment when sampling from the replay buffer, and the third concerns priority updates in PER.

Following our arrival-index convention, r_{t+1} and d_{t+1} denote the reward and terminal flag produced by taking a_t in s_t. The three issues are as follows:

1. **Incorrect trajectory-mask indexing.** When constructing `same_traj_mask` for SPR training, this line indexes the batch axis instead of the time axis. This makes every mask entry for the first sampled sequence equal to 1, even when that sequence crosses a terminal transition.
2. **Misaligned bootstrap states and terminal indicators.** The replay-based value-learning target in Eq. 1 requires rewards $r_{t+1}, \ldots, r_{t+n}$ and the bootstrap state s_{t+n}. However, this line retrieves s_{t+n-1} instead. The associated terminal check also stops at d_{t+n-1}, omitting d_{t+n} and potentially allowing bootstrapping across a terminal transition.
3. **Incorrect aggregation of replay priorities.** The loss reduction averages the per-transition losses within each minibatch before passing them to the priority update. This produces one priority per minibatch instead of one per sampled transition.

A.2 BASELINE

After correcting the implementation bugs described above, we configure a baseline agent that achieves performance comparable to SAC-BBF (RR 2). We use the following configuration changes, which were not selected through a systematic hyperparameter search:

1. **Schedule duration.** We increase T in Eq. 17 from 10K to 40K gradient updates.
2. **TD horizon.** We reduce the final n-step horizon for replay-based TD learning from 3 to 1.
3. **Entropy regularization.** We apply the exponential schedule in Eq. 17 after each parameter reset, replacing SAC-BBF's linear decay from 10^{-2} to zero over 80K environment steps.
4. **Late-stage update frequency.** We initially perform 2 gradient updates per environment step, increase this number to 4 at approximately 80K environment steps, and retain this rate for the remainder of training.

A.3 RESPRIT

ReSPRIT adds a reward head and a continuation head to the baseline. Each head is an MLP with a single hidden layer of 512 units.

The losses $\mathcal{L}_r$ and $\mathcal{L}_c$ train these prediction heads, while $\mathcal{L}_\pi^{\text{imag}}$ and $\mathcal{L}_Q^{\text{imag}}$ train the actor and action-value function on imagined trajectories. We set the corresponding loss weights to $\lambda_r = \lambda_c = 1.0$, $\lambda_\pi^{\text{imag}} = 0.1$, and $\lambda_Q^{\text{imag}} = 0.05$.

We use a shared horizon of $H = 5$ steps for SPR training and imagination. The scheduling parameter W in Eq. 15 is set to 4K gradient updates. The hyperparameter λ in the λ-return (Eq. 10) is set to 0.95, matching the value used in the DreamerV3 implementation.

A.4 EVALUATION METRICS

Let $R_{g,s}$ be the evaluation return on game g in run s, with N runs per game, and let R_g^{random} and R_g^{human} be the reference scores. We human-normalize each run as $x_{g,s} = (R_{g,s} - R_g^{\text{random}})/(R_g^{\text{human}} - R_g^{\text{random}})$. Following Agarwal et al. (2021), canonical IQM is the 25% trimmed mean of the pooled run scores $\{x_{g,s}\}$, and canonical optimality gap is the average of

Table 3: Performance point estimates and update counts on Atari 100K. BBF (RR 8) and SAC-BBF (RR 2) use published results from Schwarzer et al. (2023) and Zhang et al. (2024); game-level columns aggregate per-game means. SAC-BBF (RR 4) is reproduced using the official code. RR denotes replay ratio. Updates are approximate gradient-update counts, not comparable wall-clock costs across different objectives.

Method	IQM ↑		Mean ↑	Median ↑	Optimality gap ↓		Games > human ↑	Updates
	Canonical	Game-level			Canonical	Game-level		
Reported results								
BBF (RR 8)	1.045	1.139	2.247	0.917	0.344	0.311	12	800K
SAC-BBF (RR 2)	1.088	1.290	2.345	0.902	0.359	0.327	**13**	200K
Our results								
SAC-BBF (RR 4)	1.092	1.253	2.520	0.892	0.342	0.309	12	400K
Baseline	1.025	1.127	2.557	0.840	0.370	0.329	10	240K
ReSPRIT	**1.177**	**1.352**	**2.724**	**0.990**	**0.333**	**0.291**	**13**	240K

$\max(0, 1 - x_{g,s})$ over games and runs. The same number of runs is used for each game, so each game contributes equally to the pooled scores. Game-level metrics instead apply these operations to the per-game means $\bar{x}_g = N^{-1} \sum_{s=1}^{N} x_{g,s}$. The trimmed mean removes $\lfloor M/4 \rfloor$ observations from each tail of M sorted values; for 26 game means, this removes six from each tail. Mean and median denote the corresponding aggregates of $\{\bar{x}_g\}$, and Games > human counts games with $\bar{x}_g > 1$.

B Ablations

We examine whether imagination improves the replay-based learner and what makes that improvement possible. The first comparison uses all 26 Atari 100K games. The scheduling and loss-weight ablations each use a six-game subset selected as described in Appendix D. Each variant we evaluate uses ten independent runs per game.

B.1 Effect of imagination learning

We first test whether the compact SPR-based world model provides useful training experience beyond replay. ReSPRIT adds reward and continuation heads and imagined actor and value learning to the baseline in Appendix A.2, retaining its other modules and settings. Both agents use approximately 240K gradient updates, so the comparison measures the benefit of augmenting the existing training updates with imagination.

Table 3 shows improvements across all reported metrics. Canonical IQM rises from 1.025 to 1.177, the median from 0.840 to 0.990, and the number of games above the human reference from 10 to 13. The improvement therefore extends beyond the mean, which can be strongly influenced by a few high-scoring games. According to the raw scores in the supplementary material, ReSPRIT achieves higher mean returns than the baseline on 18 of the 26 games; ChopperCommand, KungFuMaster, and UpNDown newly exceed the human reference. These results support the central premise that SPR dynamics can contribute to behavior learning as well as representation learning.

ReSPRIT also outperforms BBF (RR 8) and SAC-BBF (RR 4) while using fewer gradient updates. Together, these comparisons show that useful additional learning can come from imagined trajectories within a strong replay-based agent.

B.2 Effect of reset-aware imagination scheduling

We test the importance of reset-aware scheduling by removing the suppression and ramp-up described in Section 3.3. The reset-agnostic variant applies imagination losses at full weight throughout training, including immediately after initialization and each reset. Following Appendix D, we evaluate on Kangaroo, PrivateEye, ChopperCommand, Breakout, RoadRunner, and Alien. The re-

Table 4: Ablation results from human-normalized scores; IQM and optimality gap use canonical aggregation. (a) Effect of reset-aware imagination scheduling on the six-game Atari 100K subset. (b) Effect of unit imagination-loss weights on a different six-game subset. Each variant uses ten independent runs per game. Baseline is the matched agent without imagination-related components. Bold marks the best result within each panel.

Method	IQM ↑	Optimality gap ↓	Mean ↑	Median ↑
ReSPRIT (reset-agnostic)	1.032	0.424	3.038	0.918
Baseline	1.157	0.440	2.930	1.016
ReSPRIT	**1.457**	**0.393**	**3.184**	**1.538**

(a) Reset-aware scheduling

Method	IQM ↑	Optimality gap ↓	Mean ↑	Median ↑
$\lambda_\pi^{\text{imag}} = 1$ $\lambda_Q^{\text{imag}} = 1$	0.444	0.544	0.821	0.392
Baseline	1.321	0.436	1.713	1.360
ReSPRIT	**1.348**	**0.429**	**1.751**	**1.670**

(b) Imagination vs. off-policy scale

ward and continuation heads are still retained across resets, and the final imagination-loss weights are unchanged. We also include the matched baseline without imagination-related components.

In Table 4(a), removing the schedule reduces IQM from 1.457 to 1.032, below the baseline's 1.157. The reset-agnostic variant retains a higher mean than the baseline, but has a lower IQM and median, indicating uneven benefits from imagination without scheduling. The raw scores in the supplementary material show a particularly clear effect on Kangaroo: removing the schedule reduces the mean return from $5{,}272.5$ to $2{,}539.5$, below the baseline's $4{,}160.2$. Breakout, however, retains a higher mean return without the schedule.

ReSPRIT performs best on all four aggregate metrics. This supports allowing the networks to readapt through replay-based learning before imagined targets regain their full influence. Retaining the prediction heads alone is insufficient: their inputs change at a reset, and the policy and value heads must also recover. Coordinating imagination with this recovery is therefore an important part of making the additional learning useful.

B.3 Effect of imagination-loss weights

We test the importance of reduced imagination-loss weights by comparing ReSPRIT's coefficients, $\lambda_\pi^{\text{imag}} = 0.1$ and $\lambda_Q^{\text{imag}} = 0.05$, with unit weights for both objectives. This increases the actor and value coefficients by factors of 10 and 20, respectively, while retaining reset-aware scheduling. Following Appendix D, we evaluate the variants on Hero, Jamesbond, Assault, Seaquest, MsPacman, and RoadRunner.

In Table 4(b), the selected weights yield an IQM of 1.348, slightly above the baseline's 1.321, whereas unit weights reduce IQM to 0.444. ReSPRIT also improves on the baseline on the other three aggregate metrics. According to the raw scores in the supplementary material, unit weights yield lower mean returns than both ReSPRIT and the baseline on all six games. On Assault, every run with unit weights scores below every ReSPRIT run, showing that the degradation on this game is not confined to a few unsuccessful runs.

The modest gain over the baseline on this subset contrasts with the large loss from increasing the weights. Imagination provides useful auxiliary supervision, but placing too much weight on its approximate returns can undermine learning from real experience. Together with the scheduling ablation, this comparison shows the importance of both when imagined targets are introduced and how strongly they influence the learner.

C Per-Game Scores Across Random Seeds

Table 5 reports ReSPRIT's evaluation returns on all 26 Atari 100K games using ten random seeds (0 through 9). Each seed entry is the mean undiscounted return over 100 evaluation episodes after 100K environment steps of training. The final column averages the ten seed-level scores for each game.

Table 5: Per-game results for ReSPRIT on Atari 100K across ten random seeds. Each seed entry reports the mean undiscounted return over 100 evaluation episodes after training for 100K environment steps. The Random and Human columns report reference scores for a random policy and human players, respectively.

			Random seed										
Game	**Random**	**Human**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Mean**
Alien	227.8	7127.7	1017.9	1676.8	1117.1	1053.3	1124.4	1153.6	1037.1	1168.5	1037.5	887.3	1127.35
Amidar	5.8	1719.5	272.39	376.76	146.96	263.56	218.55	204.83	202.08	274.72	431.83	126.22	251.79
Assault	222.4	742.0	1977.75	2941.64	2264.47	1944.1	1342.74	1350.62	1813.16	2657.61	2365.11	1572.44	2022.96
Asterix	210.0	8503.3	16737.5	12779	7767.5	18704.5	8120	18192.5	21197	40256.5	31066.5	12343	18716.40
BankHeist	14.2	753.1	779.4	571.2	78.7	1155.9	599.1	1175.5	311.1	268.9	976	993.7	690.95
BattleZone	2360.0	37187.5	9190	25070	22000	21310	23670	23450	18490	21190	23290	19990	20765.00
Boxing	0.1	12.1	92.09	82.34	88.18	80.75	50.29	58.03	91.93	83.83	89.96	90.71	80.81
Breakout	1.7	30.5	422.81	166.41	387.68	387.65	419.67	359.51	380.02	402.13	369.74	395.67	369.13
ChopperCommand	811.0	7387.8	21591	14957	2637	8678	2393	8325	3096	2853	17239	13577	9534.60
CrazyClimber	10780.5	35829.4	93156	66364	83815	68835	86678	71139	116663	74042	84529	100810	84603.10
DemonAttack	152.1	1971.0	64768.25	20088.5	48085.9	32318.45	73588.35	9669.05	21281.8	12812.35	44528.45	32023.95	35916.51
Freeway	0.0	29.6	29.52	0	30.33	31.21	29.54	0	0	31.37	0	30.23	18.22
Frostbite	65.2	4334.7	2549	2540.9	2817.7	761.4	4412.9	2996.4	3172.2	2894.9	3025	3297.4	2846.78
Gopher	257.6	2412.5	1820.2	1963.8	1707.6	2279.8	1401.8	1751.4	1129	1843.2	1574.2	2608.6	1807.96
Hero	1027.0	30826.4	6680.15	5416.3	5018.35	10333	7519.4	7458.6	7818.9	2992	6597.4	11406.35	7124.05
Jamesbond	29.0	302.8	1061	712.5	1472	1259.5	668	1539.5	853.5	363	1296	596	982.10
Kangaroo	52.0	3035.0	2014	11468	7042	8042	1872	1714	1740	7254	9863	1716	5272.50
Krull	1598.0	2665.5	6601.2	4286.7	8321.9	7820.8	7964.2	5570.1	7903.5	6747.5	7159.2	8198	7057.31
KungFuMaster	258.5	22736.3	24589	28002	12852	26334	24676	10484	14765	35902.4	24181	27902	22968.74
MsPacman	307.3	6951.6	1652	1120.9	1276.5	1712.9	1427.6	1578.7	1818.7	1736.8	1820	1558.6	1570.27
Pong	-20.7	14.6	18.96	15.57	-3.76	20.41	20.18	20.07	20.34	3.77	9.29	10.54	13.54
PrivateEye	24.9	69571.3	100	73.19	100	100	100	100	246.27	100	100	2.62	102.21
Qbert	163.9	13455.0	4559	4945.75	4388	4451.5	4764.5	4479.5	4618	4157.5	3967	4495.5	4482.63
RoadRunner	11.5	7845.0	31076	23833	27959	11724	12689	24755	31193	26268	26311	29981	24578.90
Seaquest	68.4	42054.7	972.6	1206.8	2545.7	835.4	2062.6	834.8	938.4	1076.4	2237.2	1204	1391.39
UpNDown	533.4	11693.2	13265.6	16969.1	24990.5	6882.3	6619.3	91994.9	9415.3	4621.2	13927.9	17462.1	20614.82

Table 6: Aggregate performance of ReSPRIT on Atari 100K, with 95% bootstrap confidence intervals estimated from ten runs per game.

Metric	Aggregation	Estimate	95% CI
IQM ↑	Canonical	1.177	[1.073, 1.291]
	Game-level	1.352	[1.210, 1.489]
Optimality gap ↓	Canonical	0.333	[0.313, 0.354]
	Game-level	0.291	[0.278, 0.316]
Mean ↑	—	2.724	[2.446, 3.021]
Median ↑	—	0.990	[0.863, 1.130]

C.1 Aggregate scores and uncertainty

Table 6 reports ReSPRIT's aggregate scores on Atari 100K with 95% confidence intervals, estimated by stratified bootstrapping from ten independent runs per game (Agarwal et al., 2021). The intervals summarize uncertainty in each aggregate score due to variation across runs on the 26-game benchmark.

The canonical IQM interval lies above the human-normalized reference of 1.0, supporting above-human performance under this aggregate measure. The median interval includes 1.0, reflecting uncertainty about whether the median game-level performance exceeds the human reference.

D Stratified Environment Sampling for Ablation Studies

To reduce the cost of the scheduling and loss-weight ablations, we partition the 26 Atari 100K games into three reward-density strata: sparse, medium, and dense (Table 7). These qualitative categories reflect expected reward frequency based on game mechanics and ALE environment descriptions; they are not empirical measurements of reward frequency.

Table 7: Heuristic partition of the 26 Atari 100K environments into qualitative reward-density strata used to sample environments for ablation studies.

Reward-density stratum	**Environments**
Sparse	BankHeist, BattleZone, Freeway, Hero, Jamesbond, Kangaroo, Pong, PrivateEye
Medium	Assault, Asterix, Breakout, ChopperCommand, DemonAttack, Frostbite, Gopher, Krull, Seaquest, UpNDown
Dense	Alien, Amidar, Boxing, CrazyClimber, KungFuMaster, MsPacman, Qbert, RoadRunner

Table 8: Wall-clock runtime on Asterix using the same hardware, including training and the final 100-episode evaluation. Baseline denotes the agent without imagination described in Appendix A.2.

Variant	ReSPRIT	Baseline	EZ-V2	BBF (RR 8)	DreamerV3	ITC
Runtime (hours)	2.12	1.79	6.54	6.18	4.68	12.66

We keep the partition fixed and uniformly sample two games without replacement from each stratum for each study. All variants within a study are evaluated on the same six games. We use predefined sampling seeds 0 and 1 for the scheduling and loss-weight ablations in Table 4(a) and (b), respectively, and list each sampled subset with its results.

E Computational cost

We measure wall-clock time from the start of training to the completion of the final 100-episode evaluation. All agents run on the same machine with a single NVIDIA RTX 4090 GPU, an AMD Ryzen 9 7950X3D 16-core processor, and 128 GiB of system memory. We use Asterix, which appears in EZ-V2's default Atari configuration and ITC's training example.

In Table 8, ReSPRIT takes 2.12 hours, compared with 1.79 hours for the baseline, an increase of 0.33 hours (approximately 18%) in total runtime. Both agents use approximately 240K gradient updates (Table 3); ReSPRIT adds prediction and imagination objectives within those updates. The raw scores in the supplementary material show that, on Asterix, the mean return across ten runs increases from $16{,}861.05$ to $18{,}716.40$, illustrating the trade-off between additional computation and improved learning under the same interaction budget.

ReSPRIT limits the added machinery by reusing the encoder, latent transition model, and action-value function, with short imagined rollouts and only two additional prediction heads. The added heads and imagined rollouts are used only during training; action selection still uses the learned policy without lookahead search. In the same comparison, EZ-V2, BBF (RR 8), DreamerV3, and ITC each take more than twice as long as ReSPRIT. In particular, ReSPRIT takes roughly one-third of EZ-V2's 6.54 hours. The measured runtime thus places ReSPRIT close to its replay-based baseline while providing the additional learning signals examined in Appendix B.

The comparison agents use the official repositories listed below, with their Atari 100K configurations set to Asterix and the BBF adjustments noted below.

- EZ-V2: `https://github.com/shengjiewang-jason/efficientzerov2`
- BBF: `https://github.com/google-research/google-research/tree/master/bigger_better_faster`. We set `BBFAgent.reset_every` to $5{,}000$ and `BBFAgent.replay_ratio` to 256 to match the configuration corresponding to a replay ratio of 8 used in Schwarzer et al. (2023).
- DreamerV3: `https://github.com/danijar/dreamerv3`

- ITC: official Atari 100K implementation.