Deep Active Inference Agents for Delayed and Long-Horizon Environments

Anonymous Author(s)
Affiliation
Address
email

Abstract

With the recent success of *world-model* agents—which extend the core idea of model-based reinforcement learning by learning a differentiable model for sample-efficient control across diverse tasks—*active inference* (AIF) offers a complementary, neuroscience-grounded paradigm that unifies perception, learning, and action within a single probabilistic framework powered by a generative model. Despite this promise, practical AIF agents still rely on accurate *immediate* predictions and exhaustive planning, a limitation that is exacerbated in *delayed* environments requiring planning over *long horizons*—tens to hundreds of steps. Moreover, most existing agents are evaluated on robotic or vision benchmarks which, while natural for biological agents, fall short of real-world industrial complexity. We address these limitations with a generative–policy architecture featuring (i) a *multi-step latent transition* that lets the generative model predict an entire horizon in a single look-ahead, (ii) an integrated policy network that enables the transition and receives gradients of the expected free energy, (iii) an alternating optimization scheme that updates model and policy from a replay buffer, and (iv) a single gradient step that plans over long horizons, eliminating exhaustive planning from the control loop. We evaluate our agent in an environment that mimics a realistic industrial scenario with delayed and long-horizon settings. The empirical results confirm the effectiveness of the proposed approach, demonstrating the coupled world-model with the AIF formalism yields an end-to-end probabilistic controller capable of effective decision making in delayed, long-horizon settings without handcrafted rewards or expensive planning.

1 Introduction

There has been significant progress in data-driven decision-making algorithms, particularly in reinforcement learning (RL), where agents learn policies through interaction with the environment and receive feedback [1]. Deep learning, in parallel, offers a powerful framework for extracting representations and patterns, while also enabling probabilistic modeling [2, 3], driving advancements in computer vision, natural language processing, biomedical applications, finance, and robotics. Deep RL merges these ideas—for example, by using neural function approximation in Deep Q-Networks (DQN), which achieved human-level performance on Atari games [4]. Model-based RL (MBRL) goes further by explicitly incorporating a model—either learned or provided—of the environment to guide learning and planning [5]. Similarly, the concept of world models centers on learning generative models of the environment to exploit representations and predictions of future outcomes, especially for decision-making [6]. This resonates with cognitive theories of the biological brain, which emphasize the role of internal generative models [7]. At a broader theoretical level, active inference (AIF), an emerging field in neuroscience, unifies perception, action, and learning in biological agents through the use of internal generative models [8, 9].

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

AIF is grounded in the free energy principle (FEP), which formulates neural inference and learning under uncertainty as minimization of *surprise* [10]. It provides a coherent mathematical framework that calibrates a probabilistic model governed by Bayesian inference, enabling both learning and goal-directed action directly from raw sensory inputs (i.e., *observations*) [9]. This can support the development of model-driven, adaptive agents that are trained end-to-end while offering uncertainty quantification and some interpretability [11, 12]. Similar to world models and model-based RL, AIF is powered by an internal model of the environment, which can help to capture dynamics and boost sample efficiency. Despite the potential of the AIF framework, its practical agents typically rely on accurate immediate predictions and extensive planning [12]. Such reliance can hinder performance, particularly in *delayed* environments, where the consequences of actions are not immediately observable—commonly framed in RL as *sparse rewards*, which exacerbates the credit-assignment problem [1]. Likewise, *long-horizon* tasks demand effective planning over extended temporal horizons, posing an additional challenge. These difficulties appear across diverse optimization tasks—such as manufacturing systems [11], robotics [13, 6, 14], and protein design [15, 16]—where the consequences become apparent only after many steps or upon completion of the entire process.

We explore how the potential of the AIF framework can be harnessed to build agents that remain effective in environments that are delayed and demanding long-horizon planning. Recent advances in deep generative modeling [17] have unlocked breakthroughs across diverse domains—such as AlphaFold's high-accuracy protein-structure predictions [18]. Because the generative model is the core of AIF, our objective is to extend its capacity and fidelity as the world model by predicting deep into the future. Concretely, we propose a generative model with an integrated policy network, trained end-to-end under the AIF formalism, allowing the model to produce long-horizon roll-outs and supply gradient signals to the policy during optimization. The summary of our contributions is as follows:

- We introduce an AIF-consistent generative–policy architecture that enables long-horizon predictions while providing differentiable signals to the policy.
- We derive a joint training algorithm that alternately updates the generative model and the policy network, and we show how the learned model can be leveraged during planning via gradient updates to the policy.
- We empirically demonstrate the concept's effectiveness in an industrial environment, highlighting its relevance to delayed and long-horizon scenarios.

The remainder of the paper is organized as follows: Section 2 reviews the formalism and planning strategies. Section 3 presents our proposed concept and agent architecture, while Section 4 details the experimental results. Finally, Section 5 concludes with implications and outlines future directions.

2 Background

Agents based on the world models concept extend the core idea of MBRL, learning a differentiable predictive model to facilitate policy optimization and planning via *imaginations* in the model [19, 6]. They create latent representations that capture spatial and temporal aspects to model dynamics and predict the future [19]. The architecture governing this dynamics—generative model—and how it is leveraged for policy and planning is foundational in this concept. Many designs resemble variational autoencoder [20] and are often augmented with Recurrent State-Space Models (RSSMs) to provide memory and help with credit assignment [21, 6, 14]. At the same time, RL methods such as actor–critic [1] are integrated with the model to optimize the policy [13, 6, 14], yielding sample-efficient agents that rely on imagination rather than extensive environment interaction.

AIF offers a complementary, neuroscience-grounded perspective that subsumes predictive coding that postulates that the brain minimizes prediction errors—relative to an internal generative model of the world—under uncertainty [22]. It casts the brain as a hierarchy that performs variational Bayesian inference continuously to suppress prediction error [9]. It was originally advanced to explain how organisms actively control and navigate their environments by iteratively updating beliefs and inferring actions from sensory observations [9]. AIF emphasizes the dependency of observations on actions [22]; accordingly, it posits that actions are chosen, while calibrating the model, to align with preferences and reduce uncertainty, thereby unifying perception, action, and learning [22]. The free-energy principle provides the mathematical bedrock for this framework

[23, 24], and a growing body of empirical work supports its biological plausibility [25]. AIF-based agents have been deployed in robotics, autonomous driving, and clinical decision support [26, 27, 28], demonstrating robust performance in uncertain, dynamic settings. In this work, we adopt the AIF formulation of Fountas et al. (2020) [12], which was extended in [29, 11] and has been shown to result in effective agents across different environments—such as visual and industrial tasks. We then review the planning strategies that can be coupled with this formalism.

2.1 Formalism

Within AIF, agents employ an integrated probabilistic framework consisting of an internal generative model [30] with inference mechanisms that allow them to represent and act upon the world. The framework assumes a Partially Observable Markov Decision Process [31, 30, 32], where an agent's interaction with its environment is formalized in terms of three random variables—observation, latent state, and action—denoted (o_t, s_t, a_t) at time t. In contrast to RL, this formalism does not rely on explicit reward feedback from the environment; instead, the agent learns solely from the sequence of observations it receives. The agent's generative model, parameterized by θ, is defined over trajectories as $P_\theta(o_{1:t}, s_{1:t}, a_{1:t-1})$ up to time t. The agent's behavior is driven by the imperative to minimize *surprise*, which is formulated as the negative log-evidence for the current observation, $-\log P_\theta(o_t)$ [12]. The agent approaches this imperative from two perspectives when interacting with the world, as follows [9, 12]:

1) Using the current observation, the agent calibrates its generative model by optimizing the parameters θ to yield more accurate predictions. Mathematically, this surprise can be expanded as follows [20]:

$$-\log P_\theta(o_t) \leq \mathbb{E}_{Q_\phi(s_t,a_t)}\left[\log Q_\phi(s_t, a_t) - \log P_\theta(o_t, s_t, a_t)\right] \ , \tag{1}$$

which provides an upper bound, commonly known as the negative Evidence Lower Bound (ELBO) [33]. It is widely used as a loss function for training variational autoencoders [20]. In AIF, it corresponds to the Variational Free Energy (VFE), whose minimization reduces the surprise associated with predictions relative to actual observations [12, 34, 32].

2) Looking into the future, where the agent needs to plan actions, an estimate of the surprise of future predictions can be obtained. Considering a sequence of actions—or policy—denoted as π, for $\tau \geq t$, this corresponds to $-\log P(o_\tau|\theta, \pi)$, which can be estimated analogously to VFE [35]:

$$G(\pi, \tau) = \mathbb{E}_{P(o_\tau|s_\tau,\theta)}\mathbb{E}_{Q_\phi(s_\tau,\theta|\pi)}\left[\log Q_\phi(s_\tau, \theta|\pi) - \log P(o_\tau, s_\tau, \theta|\pi)\right] \ . \tag{2}$$

This is known as the Expected Free Energy (EFE) in the framework, which quantifies the relative quality of policies—lower values correspond to better policies.

The EFE in Eq. 2 can be derived as a decomposition of distinct terms for time τ, as follows [35, 12]:

$$\begin{aligned} G(\pi, \tau) = &-\mathbb{E}_{\tilde{Q}}\left[\log P(o_\tau|\pi)\right] && \text{(3a)}\\ &+ \mathbb{E}_{\tilde{Q}}\left[\log Q(s_\tau|\pi) - \log P(s_\tau|o_\tau, \pi)\right] && \text{(3b)}\\ &+ \mathbb{E}_{\tilde{Q}}\left[\log Q(\theta|s_\tau, \pi) - \log P(\theta|s_\tau, o_\tau, \pi)\right] \ , && \text{(3c)} \end{aligned}$$

where $\tilde{Q} = Q(o_\tau, s_\tau, \theta|\pi)$. Fountas et al. (2020) [12] rearranged this formulation with further use of sampling leading to a tractable estimate for the EFE that is both interpretable and easy to calculate [12]:

$$\begin{aligned} G(\pi, \tau) = &-\mathbb{E}_{Q(\theta|\pi)Q(s_\tau|\theta,\pi)Q(o_\tau|s_\tau,\theta,\pi)}\left[\log P(o_\tau|\pi)\right] && \text{(4a)}\\ &+ \mathbb{E}_{Q(\theta|\pi)}\left[\mathbb{E}_{Q(o_\tau|\theta,\pi)}H(s_\tau|o_\tau, \pi) - H(s_\tau|\pi)\right] && \text{(4b)}\\ &+ \mathbb{E}_{Q(\theta|\pi)Q(s_\tau|\theta,\pi)}H(o_\tau|s_\tau, \theta, \pi) - \mathbb{E}_{Q(s_\tau|\pi)}H(o_\tau|s_\tau, \pi) \ . && \text{(4c)} \end{aligned}$$

Conceptually, the contribution of each term in the EFE can be interpreted as follows [12]: Extrinsic value (Eq. 4a) — the expected *surprise*, which measures the mismatch between the outcomes predicted under policy π and the agent's prior preferences over outcomes. This term is analogous to reward in RL, as it quantifies the misalignment between predicted and preferred outcomes. However, rather than maximizing cumulative reward, the agent minimizes surprise relative to preferred observations. State epistemic uncertainty (Eq. 4a) — mutual information between the agent's beliefs about states before and after obtaining new observations. This term incentivizes exploration of regions in the

environment that reduce uncertainty about latent states [12]. Parameter epistemic uncertainty (Eq. 4a) — the expected information gain about model parameters given new observations. This term also corresponds to active learning or curiosity [12], and reflects the role of model parameters θ in generating predictions. The last two terms capture distinct forms of epistemic uncertainty, providing an intrinsic drive for the agent to explore and refine its generative model. They play a role analogous to intrinsic rewards in RL that balance the exploration–exploitation trade-off. Similar information-seeking or curiosity signals underpin many successful RL algorithms—ranging from curiosity-driven bonuses [36, 37] to the entropy-regularized objective optimized by Soft Actor-Critic [38]—and have been shown to yield strong, sample-efficient agents.

2.2 Planning Strategy

Agents based on MBRL typically leverage their world model to *imagine* future trajectories before acting, trading extra computation for large gains in sample-efficiency and performance. Monte Carlo Tree Search (MCTS) [39, 40] is a notable search algorithm, which selectively explores promising trajectories in a restricted manner. Its effectiveness was highlighted in *AlphaGo Zero* [40] and later by *MuZero*, which folds a learned latent dynamics model directly into the search loop [41]. In the AIF concept, the agent's planning before taking actions is to minimize the EFE; mathematically, it corresponds to the negative accumulated EFE G as follows:

$$P(\pi) = \sigma(-G(\pi)) = \sigma\left(-\sum_{\tau>t} G(\pi,\tau)\right) , \tag{5}$$

where $\sigma(\cdot)$ represents the *Softmax* function. The agent simulates possible trajectories via roll-outs from its generative model, under policy π, to evaluate the EFE. However, calculating this any possible π is infeasible as the policy space grows exponentially with the depth of planning. Fountas et al. (2020) [12] an auxiliary module along with the MCTS to alleviate this obstacle. They proposed a recognition module [42, 43, 44], parameterized with ϕ_a as follows: *Habit*, $Q_{\phi_a}(a_t)$, which approximates the posterior distribution over actions using the prior $P(a_t)$ that is returned from the MCTS [12]. This is similar to the fast and habitual decision-making in biological agents [45]. They used this module for fast expansions of the search tree during planning, followed by calculating the EFE of the leaf nodes and backpropagating over the trajectory. Iteratively, it results in a weighted tree with memory updates for the visited nodes. They also used the Kullback–Leibler divergence between the planner's policy and the habit provides as precision to modulate the latent state [12]. They also used the Kullback–Leibler divergence between the planner's policy and the habit provides as precision to modulate the latent state [12]. Another approach to enhance the planning is using a *hybrid horizon* [11], in which the short-sighted EFE terms—based on immediate next-step predictions—are augmented with an additional term during planning to account for longer horizons. Taheri Yeganeh et al. (2024) [11] employed a Q-value network, $Q_{\phi_a}(a_t)$, to represent the amortized inference of actions, trained in a model-free manner using extrinsic values. These terms were then combined in the planner as follows:

$$P(a_t) = \gamma \cdot Q_{\phi_a}(a_t) + (1-\gamma) \cdot \sigma\left(-G(\pi)\right) , \tag{6}$$

balancing long-horizon extrinsic value against short-horizon epistemic drive.

Modern world-model agents increasingly shift the look-ahead into latent space; PlaNet [21] uses cross-entropy method roll-outs inside a RSSM trained with *latent overshooting*, while the Dreamer family [13, 6] propagates analytic value gradients through hundreds of imagined trajectories, without a tree search. EfficientZero [46] blends AlphaZero-style MCTS with latent-space imagination, surpassing human Atari performance with only 100k frames. These approaches typically couple multi-step model roll-outs with an actor (policy) and often a critic (value) network that are queried during imagination. In each simulated step, the policy proposes the next action and the critic supplies a bootstrapped value, enabling efficient multi-step look-ahead without enumerating the full action tree. Instead of sequentially sampling actions and states, Taheri Yeganeh et al . [11] trained multi-step latent transitions, conditioned on repeated actions; during planning, a single transition predicts the outcome while keeping an action for a fixed number of time-steps. This way, the impact of actions over a long horizon is captured using repeated-action simulations. While it can be combined with MCTS, this approximation helps distinguish different actions based on the EFE in highly stochastic control tasks with a single look-ahead [11]. It is limited to discrete actions, cannot go beyond repeated actions, and still requires planning via EFE computation before every action.

3 Deep Active Inference Agent

From habit-integrated MCTS to hybrid-horizon and gradient-based latent imagination, state-of-the-art agents increasingly integrate policy learning with planning to capture the long-term effects essential for scalable and sample-efficient control. A prominent approach is latent imagination, notably used by Dreamer agents [6, 21, 13], which perform sequential rollouts in latent space using a RSSM. Besides its computational burden, this method risks accumulating errors as networks are repeatedly inferred and sampled. These models embed the policy network in the latent space by sampling actions along each latent-state trajectory, so policy optimization depends on a large number of samplings in the model's imaginations.

A simpler strategy assumes a generative model that *knows* the exact form of the policy function—in other words, the network parameters themselves. We can train such a model to generate a prediction deep into the horizon with a single look-ahead, once provided with the policy parameters governing interaction with the environment over that horizon. Thus, the EFE can be computed directly over the horizon, and its gradients can be backpropagated to minimize the EFE while still guiding the agent toward its intrinsic and extrinsic objectives. Given that the policy is optimized through the gradient steps of the EFE, this approach naturally scales to continuous action space rather than choosing discrete actions, as in earlier AIF-agent implementations[12]. Here, we adopt this AIF-consistent generative-policy modeling, without incorporating further mechanisms typically employed to further enhance world models or AIF agents.

3.1 Architecture

The agent comprises, at a minimum, a **policy network** that directly interacts with the environment and a **generative model** that is trained to optimize that policy. Conditioned on the policy, the generative model constitutes the core of AIF and can be instantiated with various architectures. In this work we adopt a generic—yet commonly used—autoencoder assembly [12] to instantiate the formalism of Sec. 2.1, which requires the tightly coupled modules illustrated in Fig. **??**. Leveraging amortization [20, 43, 47] to scale inference [12], the generative model is parameterized by two sets: $\theta = \{\theta_s, \theta_o\}$ for prior generation and $\phi = \{\phi_s\}$ for recognition. Accordingly, the **Encoder** $Q_{\phi_s}(s_t)$ performs amortized inference by mapping the currently sampled observation $\tilde{o}_t$ to a posterior distribution over the latent state s_t [48]. The key difference here is that, rather than sampling actions inside the latent dynamics, we incorporate a policy function—or **Actor**—$Q_{\phi_a}(a_t \mid \tilde{o}_t)$, which itself infers a distribution over actions with parameters ϕ_a. We therefore introduce an explicit representation for the function itself with the mapping $\Pi : \mathcal{Q}_{\phi_a} \to \hat{\pi}$, resulting in $\hat{\pi}(\phi_a)$. This approach is common in neural implicit representations [49]; recent work has moreover demonstrated that neural functions with diverse computational graphs can be embedded efficiently [50]. Conditioned on the actor, the **Transition**, $P_{\theta_s}(s_{t+1} \mid \tilde{s}_t, \hat{\pi})$, *overshoots* the latent dynamics up to a planning horizon H, producing a distribution for s_{t+H} given the sampled latent state at time t, while the actor–denoted by ϕ_a–is assumed fixed throughout the horizon. Finally, the **Decoder** $P_{\theta_o}(o_{t+H} \mid \tilde{s}_{t+H})$ converts the predicted latent state back into a distribution over future observations.

Each of the three modules in the generative model is realized by a neural network that outputs the parameters of a diagonal multivariate Gaussian, thereby approximating a pre-selected likelihood family. They can be trained end-to-end by minimizing the VFE (Eq. 1), whereas the actor is optimized—using predictions from the calibrated model—by minimizing the EFE (Eq. 4). In this way, the agent unifies the two free-energy paradigms derived in the formalism. Aside from the actor and transition, which account for latent dynamics with a single look-ahead, the architecture resembles a variational autoencoder (VAE) [20]; nevertheless, other generative mechanisms, such as diffusion or memory-based RSSM models, can be extended to support the same objective.

3.2 Policy Optimization

We propose a concise yet effective formulation for embedding the actor within the generative model so that it serves as a planner that minimizes the EFE via gradient descent. Conditioned on a fixed policy $\hat{\pi}(\phi_a)$, the model generates the prediction distribution $P_\theta(o_{t+H}|\phi_a)$, from which we compute the EFE, denoted as the function $G_\theta(\tilde{o}, \phi_a)$. Policy optimization then proceeds by updating the actor parameters according to the gradient $\nabla_{\phi_a} G_\theta(\tilde{o}, \phi_a)$. Most world-model agents introduce stochasticity by sampling actions during imagination, which promotes exploration—typically aided by auxiliary

terms during the policy gradient. This results in a Monte Carlo estimation of the policy across imagined trajectories, which is then differentiated based on the return [13]. In contrast, our approach assumes the exact form of the policy is integrated into the dynamics, and exploration is driven by the AIF formalism based on the generative model.

To effectively estimate the different components of the EFE in Eq. 4, Fountas et al. (2020) [12] employed multiple levels of Monte Carlo sampling. While their original formulation incorporated sampled actions over multi-step horizons, the same structure and sampling scheme remain beneficial when using an integrated actor with deep temporal overshooting. Similarly, we adopt ancestral sampling to generate the prediction $P_\theta(o_{t+H} \mid \phi_a)$ and leverage dropout [51] in the networks. It's coupled with further sampling from the latent distributions to compute the entropies necessary for calculating the EFE terms. Crucially, under the AIF framework, agents need a prior preference over predictions to guide behavior—this is formalized through the extrinsic value (i.e., Eq. 4a). Accordingly, we define an analytical mapping that transforms the prediction distribution into a continuous preference spectrum, $\Psi : P_\theta(o_\tau) \to [0, 1]$.

Unlike RL, which relies on a monotonic return value based on accumulated rewards, this formulation allows the agent to express more general and nuanced forms of preference. In practice, designing a suitable reward function for RL agents remains a difficult task, often resulting in sparse or hand-crafted signals that can be costly to design and compute. The flexibility in preference, however, introduces challenges—particularly when agents have complex preference-space and must act with short-sighted EFE approximations. Our approach, by optimizing planning through deep temporal prediction, mitigates this issue and enables longer-term evaluation of the extrinsic value.

3.2.1 Training & Planning

During training, the generative model gradually learns how different actor parameters ϕ_a affect the dynamics, and during policy optimization, this learned dynamics is then used to differentiate the actor toward lower EFE or surprise. Critical for effective policy learning is the accuracy of the world model, which forms the foundation of AIF [23, 9, 12] and predictive coding [22]. To improve model training, we introduce experience replay [4] using a memory buffer $\mathcal{M}$, from which we sample batches of experiences, while ensuring that each batch includes the most recent transition. We compute the VFE in Eq. 1 for these experiences to train the model with β-regularization. With the updated model, we differentiate the EFE over a batch of observations—including previous and current ones—within imagined trajectories of length H, training the actor similarly to world-model methods [13, 6, 19]. This results in a joint training algorithm 1 that alternates between updating the generative model and the policy, using the model to guide planning via policy gradients. This approach, policy learning—rather than explicit action planning—relaxes the bounded-sight constraint of EFE, as the policy is iteratively trained across diverse scenarios within the planning horizon, and its effective *sight* extends beyond the nominal horizon H. Recent work on AIF-based agents has also emphasized the advantages of integrating a policy network with the EFE objective [14]. After training concludes and the agent's model is fixed, the agent can still leverage its model for planning. Specifically, EFE-based gradient updates can be applied at the observation level once every H steps, effectively fine-tuning the policy for the immediate horizon.

4 Experiments

Most existing AIF agents have shown effectiveness across a range of tasks typically performed by biological agents, such as humans and animals. These tasks often involve image-based observations [14]. For example, Fountas et al. (2020) [12] evaluated their agent on Dynamic dSprites [52] and Animal-AI [53], which biological agents can perform with relative ease. AIF has also been successfully applied in robotics [54, 29], including object manipulation [14, 27], aligning with behaviors humans naturally perform. This effectiveness is largely attributed to AIF's grounding in theories of decision-making in biological brains [9]. However, applying AIF to more complex domains—such as industrial system control—poses significant challenges. Even humans may struggle to design effective policies in these settings. Such environments often exhibit high stochasticity, where short observation trajectories are dominated by noise, making it difficult to optimize free energy for learning and action selection. This issue is less pronounced in world model agents, which often use memory-based (e.g., recurrent) architectures [13, 6]. Moreover, realistic environments frequently combine discrete and continuous observation modalities, complicating generative and

Algorithm 1 Deep AIF Agent Training (per epoch)

Initialize $\theta = \{\theta_s, \theta_o\},\ \phi = \{\phi_s, \phi_a\},\ \mathcal{M}$
Randomly initialize E
for $n = 1, 2, ..., N$ **do**
  $\hat{\pi}_t \leftarrow \Pi(\mathcal{Q}_{\phi_a})$
  for $\tau = t+1, t+2, ..., t+H$ **do**
    Sample a new observation $\tilde{o}_\tau$ from E
    Apply $\tilde{a}_\tau \sim Q_{\phi_a}(a_\tau|\tilde{o}_\tau)$ to E
    Sample a new observation $\tilde{o}_{\tau+1}$ from E
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{(\tilde{o}_t, \hat{\pi}_t, \tilde{o}_{t+H})\}$
  $\{(\tilde{o}_{t'}, \hat{\pi}_{t'}, \tilde{o}_{t'+H})\} \sim \mathcal{M}$
  for $t' = 1, 2, ..., B_m$ **do**
    run Model$(\tilde{o}_{t'}, \hat{\pi}_{t'}, \tilde{o}_{t'+H})$
    $\mathcal{L}_s \leftarrow \mathcal{L}_s + D_{\mathrm{KL}}\left[Q_{\phi_s}(s_{t'+H}) \,||\, \mathcal{N}(\mu, \sigma^2)\right]$
    $\mathcal{L}_o \leftarrow \mathcal{L}_o - \mathbb{E}_{Q(s_{t'+H})}\left[\log P_{\theta_o}(o_{t'+H}|\tilde{s}_{t'+H})\right]$
    $\mathcal{L}_o \leftarrow \mathcal{L}_o + \beta * D_{\mathrm{KL}}\left[Q_{\phi_s}(s_{t'+H}) \,||\, \mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2)\right]$
  $\theta_s \leftarrow \theta_s - \xi \nabla_{\theta_s} \mathbb{E}\big[\mathcal{L}_s(\theta_s)\big]$
  $\phi_s \leftarrow \phi_s - \gamma \nabla_{\phi_s} \mathbb{E}\big[\mathcal{L}_s(\phi_o)\big]$
  $\theta_o \leftarrow \theta_o - \eta \nabla_{\theta_o} \mathbb{E}\big[\mathcal{L}_o(\theta_o)\big]$
  for $\tau = 1, 2, ..., B_a$ **do**
    $\{\tilde{o}_\tau\} \sim \mathcal{M}$
    Compute $Q_{\phi_s}(s_\tau)$ using $\tilde{o}_\tau$
    Sample $\tilde{s}_\tau \sim Q_{\phi_s}(s_\tau)$
    for $s = 1, 2, ..., S$ **do**
      Compute μ, σ from $P_{\theta_s}(s_{\tau+H}|\tilde{s}_\tau, \hat{\pi}_t)$
      Sample $\tilde{s}_{\tau+H} \sim \mathcal{N}(\mu, \sigma^2)$
      Compute $P_{\theta_o}(o_{\tau+H}|\tilde{s}_{\tau+H})$
      Compute $Q_{\phi_s}(\tilde{s}_{\tau+H})$ using $\tilde{o}_{\tau+H}$
      Compute $\mu', \sigma' \leftarrow Q_{\phi_s}(\tilde{s}_{\tau+H})$
      $G \leftarrow G + \Phi(P_{\theta_o}(o_{\tau+H}|\tilde{s}_{\tau+H}))$
      $G \leftarrow G + [H(\mu', \sigma') - H(\mu, \sigma)]$
    for $s = 1, 2, ..., S$ **do**
      Sample $\tilde{s}_{\tau+H} \sim P_{\theta_s}(s_{\tau+H}|\tilde{s}_\tau, \hat{\pi}_\tau)$ ▷ *Re-computed with dropout.*
      Compute $\mu'',\ \sigma'' \leftarrow P_{\theta_o}(o_{\tau+H}|\tilde{s}_{\tau+H})$
      Sample $\tilde{s}_{\tau+H} \sim \mathcal{N}(\mu, \sigma^2)$
      Compute $\mu''',\ \sigma''' \leftarrow P_{\theta_o}(o_{\tau+H}|\tilde{s}_{\tau+H})$
      $G \leftarrow G + [H(\mu'',\ \sigma'') - H(\mu''',\ \sigma''')]$
  $\phi_a \leftarrow \phi_a - \alpha \nabla_{\phi_a} \mathbb{E}\big[G(\phi_a)\big]$

Agent components:
- Model:
  - Encoder Q_{ϕ_s}.
  - Transition P_{θ_s}.
  - Decoder P_{θ_o}.
- Actor Q_{ϕ_a}.
- Actor mapping Π.
- Preference mapping Ψ.

Other components:
- Environment E.
- Memory buffer $\mathcal{M}$.

Hyperparameters:
- Iterations N.
- Beta β.
- Horizon H.
- Batch size B_m , B_a.
- Sample size S.
- Learning rate ξ , γ , η , α.

Run Model$(\tilde{o}_i, \hat{\pi}, \tilde{o}_{i+H})$:
- Compute $Q_{\phi_s}(s_i)$ using $\tilde{o}_i$
- Sample $\tilde{s}_i \sim Q_{\phi_s}(s_i)$
- Compute $\mu, \sigma \leftarrow P_{\theta_s}(s_{i+H}|\tilde{s}_i, \hat{\pi})$
- Compute $Q_{\phi_s}(\tilde{s}_{i+H})$ using $\tilde{o}_{i+H}$
- Compute $\mu', \sigma' \leftarrow Q_{\phi_s}(\tilde{s}_{i+H})$
- Sample $\tilde{s}_{i+H} \sim \mathcal{N}(\mu, \sigma^2)$
- Compute $P_{\theta_o}(o_{i+H}|\tilde{s}_{i+H})$

sampling predictions. Delayed feedback and long-horizon requirements further challenge planning under the AIF framework. Additionally, many real-world tasks require rapid, frequent decisions and sustained performance in non-episodic, stochastic settings. To assess our approach, we employ a high-fidelity simulation environment validated to reflect realistic industrial control scenarios [55], which incorporates all the above challenges [11].

4.1 Application

We focus on simulating workstations in an automotive manufacturing system composed of parallel, identical machines (see Appendix for details). As energy efficiency becomes increasingly critical in manufacturing [56], RL offers a model-free alternative to traditional control, though it may struggle with rapid adaptations in non-stationary environments [57]. Governed by Poisson processes for arrivals, processing, failures, and repairs [55], the system evolves as a discrete-time Markov chain [58]. Control actions—switching machines on or off—aim to improve energy efficiency without compromising throughput. Due to stochastic delays, the system connects continuous-time dynamics to discrete-time decisions, making performance only observable over long horizons. Accordingly,

we employ a window-based preference metric [11] that evaluates KPIs over the past eight hours. The production rate is defined as $T = \frac{N(t)-N(t-t_s)}{t_s}$, where $N(t)$ is the number of parts produced, and the energy consumption rate as $E = \frac{C(t)-C(t-t_s)}{t_s}$, where $C(t)$ denotes total energy consumed, with $t - t_s \approx 8$ hrs. This window may span thousands of actions, where due to stochasticity and the integral nature of performance, immediate observations are noisy and uninformative. As a result, the AIF agents based on short-horizon EFE planning are not feasible in this setting. By operating directly on raw performance signals rather than handcrafted rewards, the approach enables scalability to domains where reward signals are sparse or expensive. The agent must handle delayed feedback and plan over extended horizons to move towards the preferred performance. This problem is continual with no terminal state, and decisions rely on both discrete and continuous observations.

4.2 Results

To validate the performance of our agent in the aforementioned environment, we adopted a rigorous evaluation scheme based on Algorithm 1. Unlike previous works that used batch interactions to improve training efficiency [12], our agent was trained in each epoch by interacting with a single environment instance, reflecting a more challenging setting. The trained agent's performance was then evaluated across several randomly initialized environments. From these, the best-performing instance was selected for a one-month simulation run to assess energy efficiency and production loss, in comparison to a baseline scenario where no control was applied and machines were continuously active. We also constructed a compositional preference score—analogous to a reward function—based on time-window KPIs for energy consumption and production, serving as an overall indicator of agent performance, which is part of the observation of the agent. To enforce further regularization in the latent space to match a normal distribution, we used a *Sigmoid* function in its non-saturated domain. Since we needed to encode the actor function, which is essentially a computational graph [50], we adopted a simple, non-parametric mapping Π that concatenates the input with the first hidden and output values. Given its input-output structure and the fact that the model was continuously trained with that, this mapping effectively serves as an approximation of the actor's neural function (see Appendix for details on the agent and experimental setup).

We implemented the agent in the exact production system, using parameters verified to reflect realistic conditions, following the aforementioned scheme. Figure 1 presents the performance of the agent with an overshooting horizon of $H = 300$. During evaluations after each epoch (100 iterations), the agent improved the preference score of observations (Fig. 1a), which correlates with increased energy efficiency (Fig. 1b). Notably, the EEF of imagined trajectories used for policy updates decreased as the agent learned to control the system. This trend is observed in both the extrinsic and uncertainty components of the EFE. Since policy optimization relies heavily on learning a robust generative model—with the actor integrated within it—the agent gradually improved its predictive capacity and reduced reconstruction error across both discrete (Fig. 1d, preference) and continuous (Fig. 1e,f, machine and buffer states) elements of the observation space. While EFE and overall performance eventually stabilized, the generative model continued to improve, indicating that full reconstruction of future observations is not strictly required for effective control. The agent manages to improve the performance even when the overshooting horizon can be longer (e.g., $H = 1000$ steps; see Appendix). We then evaluated the trained agent over one month of simulated interaction (10 replications), applying gradient updates every H steps during planning. Loffredo et al. (2023) [57] tested model-free RL agents across a reward parameter ϕ , with DQN emerging as the top performer. Table1 shows that our DAIF agent outstrips this baseline, raising energy efficiency per production unit by $10.21\% \pm 0.14\%$ while keeping throughput loss negligible.

5 Conclusion and Future Work

We introduced *Deep Active Inference Agents* (DAIF) that integrate a multi-step latent transition and an explicit, differentiable policy inside a single generative model. By overshooting the dynamics to a long horizon and back-propagating expected-free-energy gradients into the policy, the agent plans without an exhaustive tree search, scales naturally to continuous actions, and preserves the epistemic–exploitative balance that drives active inference. We evaluated DAIF on a high-fidelity industrial control problem whose feature complexity has rarely been tackled in previous works based on active inference. Empirically, DAIF closed the loop between model learning and control in highly

Agent(ϕ)	Production Loss [%]	EN Saving [%]
DQN (0.93)	4.82 ± 0.34	10.87 ± 0.76
DQN (0.94	3.34 ± 0.23	9.92 ± 0.69
DAIF	$\mathbf{2.59 \pm 0.16}$	$\mathbf{12.49 \pm 0.04}$
DQN (0.95)	1.27 ± 0.05	7.00 ± 0.07
DQN (0.96)	1.27 ± 0.09	7.62 ± 0.12
DQN (0.97)	1.20 ± 0.05	7.72 ± 0.10
DQN (0.98)	0.54 ± 0.04	2.72 ± 0.19
DQN (0.99)	0.40 ± 0.03	2.46 ± 0.01

Table 1: Production loss versus energy-saving (EN) across reward parameters ϕ of DQN agents [57] and for the DAIF agent.

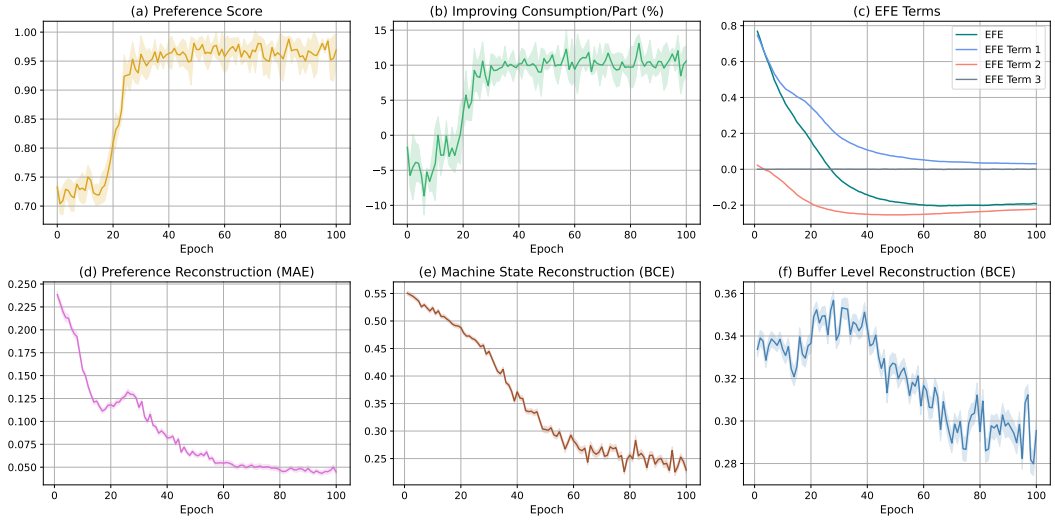

Figure 1: The performance of the agent with $H = 100$ on the real industrial system.

stochastic, delayed, long-horizon environment. With a single gradient update every H steps, the trained agent planed, and achieved strong performance—surpassing model-free RL baseline—while its world model continued to refine predictive accuracy even after the policy stabilized.

Limitations and future work: While predicting an H-step transition removes the expensive *per-step* planning loop, the agent still has to gather *experience* after H interactions and store it in the replay buffer for training, so its sample-efficiency can still be improved. To update the world model after each new environment interaction—obtained under different actor/moving parameters—we need an operator that aggregates the *sequence* of actor representations. Recurrent models are a natural choice for this, but their sequential unrolling adds latency and can hinder gradient flow. A lighter alternative is to treat the H embeddings as an (almost) unordered set and use a set function [59]; when the temporal structure with simple positional embeddings (e.g. sinusoidal [60]) can be concatenated before the set pooling. This allows us to break the horizon into segments—down to a single step—and still backpropagate EFE gradients during planning through the aggregations the current policy representation. Finally, (neural) operator-learning techniques could enable resolution-invariant aggregation across function spaces [61, 62]. Additional extensions include replacing the VAE world model with diffusion- or flow-matching-based generators [28], adopting actor–critic optimization (as in Dreamer and related world-model agents [13, 6, 14]), and introducing regularization schemes to stabilize EFE gradient updates and reduce their variance. Rapid adaptation in non-stationary settings—where model-free agents often struggle—remains an especially promising direction.

Overall, this work bridges neuroscience-inspired active inference and contemporary world-model RL, demonstrating that a compact, end-to-end probabilistic agent can deliver efficient control in domains where hand-crafted rewards and step-wise planning are impractical.

References

[1] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[3] Christopher M. Bishop and Hugh Bishop. *Deep Learning: Foundations and Concepts*. Springer International Publishing, 2024.

[4] 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.

[5] Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

[6] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025.

[7] Karl Friston, Rosalyn J. Moran, Yukie Nagai, Tadahiro Taniguchi, Hiroaki Gomi, and Joshua B. Tenenbaum. World model learning and inference. *Neural Networks*, 144:573–590, 2021.

[8] Karl Friston, Thomas FitzGerald, Francesco Rigoli, Philipp Schwartenbeck, and Giovanni Pezzulo. Active inference: a process theory. *Neural computation*, 29(1):1–49, 2017.

[9] Thomas Parr, Giovanni Pezzulo, and Karl J Friston. *Active inference: the free energy principle in mind, brain, and behavior*. MIT Press, 2022.

[10] Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138, 2010.

[11] Yavar Taheri Yeganeh, Mohsen Jafari, and Andrea Matta. Active inference meeting energy-efficient control of parallel and identical machines. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 479–493. Springer, 2024.

[12] Z. Fountas, Noor Sajid, Pedro A. M. Mediano, and Karl J. Friston. Deep active inference agents using monte-carlo methods. *ArXiv*, abs/2006.04176, 2020.

[13] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[14] Viet Dung Nguyen, Zhizhuo Yang, Christopher L Buckley, and Alexander Ororbia. R-aif: Solving sparse-reward robotic tasks from pixels with active inference and world models. *arXiv preprint arXiv:2409.14216*, 2024.

[15] Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. In *International conference on learning representations*, 2019.

[16] Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Tommaso Biancalani, Avantika Lal, Tommi Jaakkola, Sergey Levine, Hanchen Wang, and Aviv Regev. Fine-tuning discrete diffusion models via reward optimization with applications to dna and protein design. *arXiv preprint arXiv:2410.13643*, 2024.

[17] Jakub M Tomczak. *Deep Generative Modeling*. Springer Cham, 2024.

[18] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia

Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M.R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

[19] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[21] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[22] Beren Millidge, Tommaso Salvatori, Yuhang Song, Rafal Bogacz, and Thomas Lukasiewicz. Predictive coding: towards a future of deep learning beyond backpropagation? *arXiv preprint arXiv:2202.09467*, 2022.

[23] Karl Friston, Francesco Rigoli, Dimitri Ognibene, Christoph Mathys, Thomas Fitzgerald, and Giovanni Pezzulo. Action and behavior: A free-energy formulation. *Biological Cybernetics*, 102(3):227–260, 2010.

[24] Beren Millidge. Applications of the free energy principle to machine learning and neuroscience. *arXiv preprint arXiv:2107.00140*, 2021.

[25] Takuya Isomura, Kiyoshi Kotani, Yasuhiko Jimbo, and Karl J Friston. Experimental validation of the free-energy principle with in vitro neural networks. *Nature Communications*, 14(1):4547, 2023.

[26] Corrado Pezzato, Carlos Hernández Corbato, Stefan Bonhof, and Martijn Wisse. Active inference and behavior trees for reactive action planning and execution in robotics. *IEEE Transactions on Robotics*, 39(2):1050–1069, 2023.

[27] Tim Schneider, Boris Belousov, Georgia Chalvatzaki, Diego Romeres, Devesh K Jha, and Jan Peters. Active exploration for robotic manipulation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9355–9362. IEEE, 2022.

[28] Yufei Huang, Yulin Li, Andrea Matta, and Mohsen Jafari. Navigating autonomous vehicle on unmarked roads with diffusion-based motion prediction and active inference. *arXiv preprint arXiv:2406.00211*, 2024.

[29] Lancelot Da Costa, Pablo Lanillos, Noor Sajid, Karl Friston, and Shujhat Khan. How active inference could help revolutionise robotics. *Entropy*, 24(3):361, 2022.

[30] Lancelot Da Costa, Noor Sajid, Thomas Parr, Karl Friston, and Ryan Smith. Reward maximization through discrete active inference. *Neural Computation*, 35(5):807–852, 2023.

[31] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[32] Aswin Paul, Noor Sajid, Lancelot Da Costa, and Adeel Razi. On efficient computation in active inference. *arXiv preprint arXiv:2307.00504*, 2023.

[33] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[34] Noor Sajid, Francesco Faccio, Lancelot Da Costa, Thomas Parr, Jürgen Schmidhuber, and Karl Friston. Bayesian brains and the rényi divergence. *Neural Computation*, 34(4):829–855, 2022.

[35] Philipp Schwartenbeck, Johannes Passecker, Tobias U Hauser, Thomas HB FitzGerald, Martin Kronbichler, and Karl J Friston. Computational mechanisms of curiosity and goal-directed exploration. *elife*, 8:e41703, 2019.

[36] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[37] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[38] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[39] Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pages 72–83. Springer, 2006.

[40] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[41] 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.

[42] Alexandre Piché, Valentin Thomas, Cyril Ibrahim, Yoshua Bengio, and Chris Pal. Probabilistic planning with sequential monte carlo methods. In *International Conference on Learning Representations*, 2018.

[43] Joe Marino, Yisong Yue, and Stephan Mandt. Iterative amortized inference. In *International Conference on Machine Learning*, pages 3403–3412. PMLR, 2018.

[44] Alexander Tschantz, Beren Millidge, Anil K Seth, and Christopher L Buckley. Control as hybrid inference. *arXiv preprint arXiv:2007.05838*, 2020.

[45] Matthijs Van Der Meer, Zeb Kurth-Nelson, and A David Redish. Information processing in decision-making systems. *The Neuroscientist*, 18(4):342–359, 2012.

[46] Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Advances in neural information processing systems*, 34:25476–25488, 2021.

[47] Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014.

[48] Charles C Margossian and David M Blei. Amortized variational inference: When and why? *arXiv preprint arXiv:2307.11018*, 2023.

[49] Emilien Dupont, Hyunjik Kim, SM Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. *arXiv preprint arXiv:2201.12204*, 2022.

[50] Miltiadis Kofinas, Boris Knyazev, Yan Zhang, Yunlu Chen, Gertjan J Burghouts, Efstratios Gavves, Cees GM Snoek, and David W Zhang. Graph neural networks for learning equivariant representations of neural networks. *arXiv preprint arXiv:2403.12143*, 2024.

[51] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[52] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2016.

[53] Matthew Crosby, Benjamin Beyret, and Marta Halina. The animal-ai olympics. *Nature Machine Intelligence*, 1(5):257–257, 2019.

[54] Pablo Lanillos, Cristian Meo, Corrado Pezzato, Ajith Anil Meera, Mohamed Baioumy, Wataru Ohata, Alexander Tschantz, Beren Millidge, Martijn Wisse, Christopher L Buckley, et al. Active inference in robotics and artificial agents: Survey and challenges. *arXiv preprint arXiv:2112.01871*, 2021.

[55] Alberto Loffredo, Marvin Carl May, Louis Schäfer, Andrea Matta, and Gisela Lanza. Reinforcement learning for energy-efficient control of parallel and identical machines. *CIRP Journal of Manufacturing Science and Technology*, 44:91–103, 2023.

[56] Alberto Loffredo, Nicla Frigerio, Ettore Lanzarone, and Andrea Matta. Energy-efficient control in multi-stage production lines with parallel machine workstations and production constraints. *IISE Transactions*, 56(1):69–83, 2024.

[57] Alberto Loffredo, Marvin Carl May, Andrea Matta, and Gisela Lanza. Reinforcement learning for sustainability enhancement of production lines. *Journal of Intelligent Manufacturing*, pages 1–17, 2023.

[58] Sheldon M Ross. *Introduction to probability models*. Academic press, 2014.

[59] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[61] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[62] Lu Lu, Pengzhan Jin, Giovanni Pang, Zhiping Zhang, and George Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

1. **Claims**

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

   Answer: [Yes]

   Justification: The four claims made in the abstract and introduction are the main contributions and are substantiated throughout the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed in the final section of the paper, outlining areas for improvement and directions for future work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

Justification: The paper does not present new theoretical results or formal theorems. However, relevant assumptions are clearly stated and referenced where applicable.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper provides sufficient details in the Appendix to reproduce the main experimental results, including architecture definitions, training and evaluation procedures, environment description, preference mapping, and hyperparameters.

Guidelines:

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

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

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

Answer: [Yes]

Justification: We provide anonymized access to the code and training scripts via a link in the supplementary material. Instructions are included to replicate the environment, run the training pipeline, and evaluate performance.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: The appendix includes detailed descriptions of the experimental setup to produce the reported results.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Error bars representing one standard deviation over 10 random seeds are reported in all main result tables and plots. We provide complete information in appendix.

Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper provides this information in the appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research adheres fully to the NeurIPS Code of Ethics.

   Guidelines: The research adheres fully to the NeurIPS Code of Ethics.

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper presents a method for planning in industrial control via deep active inference, with no direct societal deployment or sensitive data usage. As a foundational contribution focused on simulation-based optimization, it does not raise immediate societal concerns.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper does not release models or data with high risk for misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All external code or datasets used in the project are properly cited in the paper, and their licenses are respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

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

13. **New assets**

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

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

    Justification: The research does not involve human participants or any form of crowdsourcing.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [NA]

    Justification: The study does not involve human subjects and thus does not require IRB approval.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: No large language models (LLMs) were used in the development of the core methodology.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.