Reinforcing Good Experiences: AWAC-DreamerV3 with Confidence-Aware Policy Gating

強化好經驗: 具信心感知策略閘控的優勢權重演員評論家-DreamerV3

專題領域：資工領域

組　　別：B559

指導教授：吳尚鴻 教授

組員姓名：黃祥陞 (Huang, Hsiang-Sheng)、江承瀚(Chiang, Cheng-Han)

研究期間：114 年 1 月 1 日至 114 年 11 月 10 止，共 10 個月

Abstract

We aim to collect and reinforce good experiences by integrating Advantage-Weighted Actor-Critic (AWAC) into DreamerV3. Our approach adds a second policy network trained using AWAC to emphasize high-value transitions from replay data. Because DreamerV3 relies heavily on imagined rollouts that can drift when the learned dynamics are imperfect, the AWAC head serves as a grounding mechanism that amplifies reliable behavior directly from real replay.

To intelligently choose between the original Dreamer policy and the AWAC policy, we introduce a gating mechanism that evaluates policy confidence through normalized entropy and measures distributional agreement via symmetric KL divergence. The gate dynamically selects the more confident policy while being conservative when policies disagree significantly. This dual-policy architecture with intelligent gating enables the system to effectively leverage both the robust world-model learning of DreamerV3 and the experience-focused learning of AWAC. Our method provides formal analysis of gate properties including boundedness and monotonicity. Experiments on DMC proprioceptive control and Crafter demonstrate stable gating behavior and competitive sample efficiency, showing effective reinforcement of good experiences while maintaining robust performance.

Our method opens up new possibilities for **lifelong learning** by letting **DreamerV3** handle open-ended internal **imagination**, while **AWAC** focuses on consolidating high-quality **experiences** from real interactions. This complementary "imagination × experience" dynamic allows the agent to explore broadly yet learn stably from its past. With the dual-policy architecture and gating mechanism, the system can continually accumulate useful behaviors, reduce **catastrophic forgetting**, and maintain robust, long-term adaptability in evolving environments.

摘要

我們的目標是將 **優勢權重演員–評論家（Advantage-Weighted Actor-Critic, AWAC）**整合進 **DreamerV3**，以更有效地蒐集並強化優良經驗。我們在架構中加入第二個以 **AWAC** 訓練的策略網路，專門從重播資料中突出高價值轉換。由於 **DreamerV3** 嚴重依賴想像軌跡（imagined rollouts），在學得的動態模型不完美時容易產生偏移，我們藉由 **AWAC head** 提供「扎根（grounding）」機制，直接從真實重播資料中放大可靠行為。

為了在原始 **Dreamer** 策略與 **AWAC** 策略之間智能切換，我們設計了一個 **閘門機制（gating mechanism）**：一方面利用 **正規化熵（normalized entropy）** 來估計策略的信心，另一方面透過 **對稱 KL 散度（symmetric KL divergence）** 來衡量兩者機率分佈的一致程度。當兩個策略分佈差異過大時，閘門會採取較為保守的決策；在一般情況下，則動態選擇更具信心的一方。

這樣的雙策略（dual-policy）架構結合了 **DreamerV3** 在世界模型學習上的穩健性，以及 **AWAC** 對真實經驗的聚焦強化能力。針對閘門，我們也提供其性質的形式化分析，包括有界性與單調性。在 **DMC 本體感知控制（proprioceptive control）** 與 **Crafter** 的實驗結果顯示，我們的方法具備穩定的閘門行為與具有競爭力的取樣效率，能在強化良好經驗的同時，維持整體表現的穩健性。

進一步而言，我們的方法為 **終身學習（lifelong learning）** 帶來了新的可能性。透過 **DreamerV3** 進行「想像（imagination）」，得以不斷在內部世界拓展理解與推理能力；同時 **AWAC** 則學習了「經驗 (Experiences)」，專注於真實互動中累積與強化值得保留的行為。「想像 × 經驗」的互補動態，讓代理人既能勇於探索，也能穩定地從歷史中學習。

在這樣的雙策略協同下，我們的架構能隨著時間持續吸收優良行為、抑制錯誤偏移，並透過閘門機制在不同任務階段做出最可靠的策略選擇。這不僅減少了長期學習中常見的 **災難性遺忘（catastrophic forgetting）**，也為代理人(Agent)在多階段任務與長期演化環境中維持穩定成長奠定基礎。因此，結合 DreamerV3 的想像力與 AWAC 的經驗累積能力，我們的方法展現出在終身學習場景中持續精進與永續適應的潛在優勢。

目錄

1. Introduction & Background

世界模型（World Model）是強化學習中一類能自主學習環境動態的生成模型，透過壓縮歷史觀測、動作與獎勵訊息，使智能體得以在內部構建對環境的可預測表徵。藉由這個內在模型，**智能體可以在潛在空間中模擬未來軌跡、評估行動後果**，進而在無需真實互動的情況下進行大規模**「想像式（imagination-based）」**策略學習。基於模型的強化學習（model-based reinforcement learning）正是透過學習這類能捕捉環境動態的世界模型（world model）來提升樣本效率（sample efficiency），使得**智能體可以更多依賴「想像（imagination）」**而非昂貴且可能具風險的真實互動來學習策略 [1]，因此在資料收集成本高或具風險的情境中特別具有優勢。

DreamerV3 [2] 代表了世界模型式演算法中的最新進展，在固定超參數設定下便能在多種領域展現優異表現，省去了大量調整超參數的成本。該演算法透過學習 **遞迴狀態空間模型（Recurrent State-Space Model, RSSM）來刻畫環境動態，並同時利用想像軌跡與重播資料（replay data）訓練演員–評論家（actor-critic）**策略。這種結合「想像規劃（imagination-based planning）」與「重播學習（replay-based learning）」的雙重學習架構，使 DreamerV3 能在連續與離散控制任務、視覺與本體感知（proprioceptive）輸入，以及多樣化獎勵結構的環境中展現強健的泛化能力。

儘管 DreamerV3 已展現出優異的整體性能，我們進一步探索在世界模型架構下，如何更有效「收集並強化良好經驗」。在不改動核心世界模型學習機制的前提下，我們聚焦於：如何更好地運用重播資料中那些正向、有利的經驗來強化策略學習。這一方向建立在一個關鍵洞見之上：若能更強調相對高價值的轉換（high-value transitions），便有機會加速學習並進一步提升樣本效率。

我們觀察到這個分層學習的架構，會有以下的問題: **當學得的動態模型仍不精確時，DreamerV3 所產生的想像軌跡可能會偏離真實環境。這類模型偏差（model bias）可能導致演員過度擬合於「幻覺式（hallucinated）」的想像軌跡，尤其是在訓練早期階**

段。因此，我們希望引入一個**以重播為基礎的訊號（replay-grounded signal）**，能持續地將策略推向在真實環境中「確實有效」的行為。**優勢加權演員–評論家（Advantage-Weighted Actor-Critic, AWAC）**[3] 正好具備這項特性：它從真實經驗中均勻取樣，並對「出乎意料地好」的轉換進行指數加權，形成一個高信度的校正訊號，與想像過程互相補充。

為了達成上述目標，我們提出一種將 **AWAC** 與 **DreamerV3** 相結合的方法。我們在架構中加入第二個策略網路，透過**優勢加權回歸（advantage-weighted regression）在重播資料上進行訓練，以強調其中具價值的經驗。為了在原始 Dreamer 策略與 AWAC 策略之間做出智慧選擇，我們設計了一個同時評估策略「信心」與「一致性」的閘門機制（gating mechanism）**。這個雙策略（dual-policy）架構，使系統能同時受惠於 DreamerV3 穩健的世界模型學習，以及 AWAC 專注於真實經驗的學習優勢。

本研究的主要貢獻如下：

(1) 提出一個整合 AWAC 與 DreamerV3 的雙策略架構，透過優勢加權學習來收集並強化良好經驗，補足模型的想像與實際的偏差問題。

(2) 設計一個根據策略信心與分佈一致性進行選擇的智慧閘門機制，並對其性質（包含有界性與單調性）進行形式化分析。

2. Related Work

2.1 DreamerV3

DreamerV3 [2] 是一個由 **兩個主要部分** 所構成的世界模型（world model）：**遞迴狀態空間模型（Recurrent State-Space Model, RSSM）** 與 **演員–評論家（actor-critic）網路**。（見 Fig 1-1）

2.1.1 遞迴狀態機 Recurrent State Space Model (RSSM)

RSSM 提供了一個「虛擬環境」，其中包含用於預測觀測、獎勵、續存（continuation）與價值的多個預測頭（predictive heads）。這讓演員–評論家網路能在訓練期間於「潛空間（latent space）」中進行想像軌跡（imagined rollouts）的學習。（見 Fig 1-2,1-3）

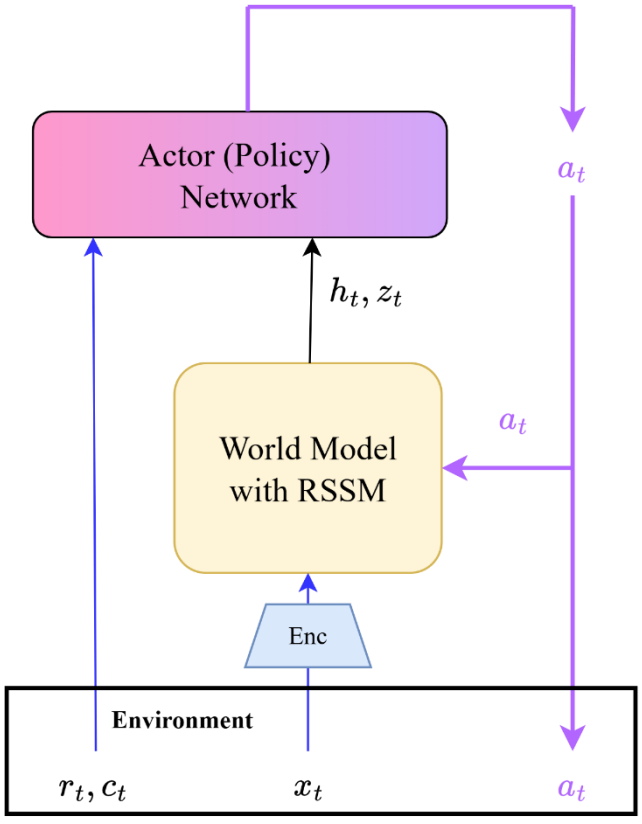


Fig. 1-1 DreamerV3做推論的架構

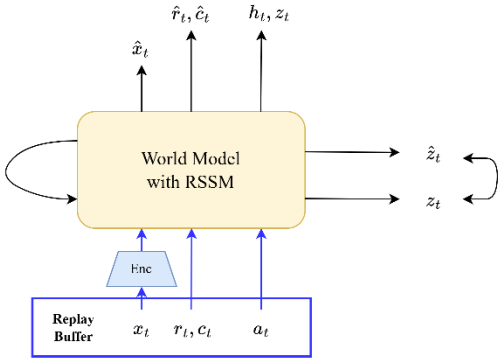


Fig. 1-2 DreamerV3中的 RSSM 學習方式，用於創造潛在空間，可以在此環境 Actor-Critic。且 RSSM 只從 Replay Buffer Offline 訓練。

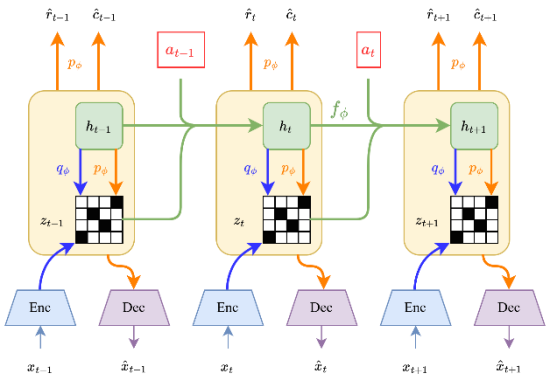


Fig. 1-3 DreamerV3中 RSSM 內部的架構，包含了3種 function: p_ϕ , q_ϕ , f_ϕ 這3種 function 可以訓練出一個潛在空間。

我們的 RSSM 實作遵循原始論文的定義：

$$\text{RSSM} \quad \begin{cases} h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1}) & (Sequence) \\ z_t \sim q_\phi(z_t | h_t, x_t) & (Encoder) \\ \widehat{z_t} \sim p_\phi(\widehat{z_t} | h_t) & (Dynamics) \end{cases}$$

$$\widehat{r_t} \sim p_\phi(\widehat{r_t} | h_t, z_t) \quad (Reward)$$
$$\widehat{c_t} \sim p_\phi(\widehat{c_t} | h_t, z_t) \quad (Continue)$$
$$\widehat{x_t} \sim p_\phi(\widehat{x_t} | h_t, z_t) \quad (Decoder)$$

RSSM 有幾個主要的函數:

1. 預測性函數 Predictive Function: p_ϕ ，預測 $\widehat{z_t}, \widehat{r_t}, \widehat{c_t}, \widehat{x_t}$

預測性函數旨在預測環境的輸出，例如環境本身($\widehat{x_t}$), 持續信號($\widehat{c_t}$) 以及獎勵信號 $\widehat{r_t}$ ；但也會預測潛在空間的動態$\widehat{z_t}$，預測潛在空間的表徵。 (式2)

2. 特徵嵌入器/編碼器 Encoder Function: q_ϕ ，用於將長期記憶及輸入表徵嵌入為z_t。

3. 序列性函數 Sequence Function: f_ϕ ，用於將經驗變成長期記憶。

這3個函數可以讓預測性函數預測的$\widehat{z_t}$ 與實際環境編碼後給出的 z_t 是對齊 (align) 的，並使用 KL-divergence 拉近兩個 distribution (式3, 4) ，並且回傳 gradient 給 h_t ，使 Sequence Function 也可以做 learning。 另外z_t 採用 vector quantized 的 one hot 設計。

$$\mathcal{L}(\phi) \doteq E_{q_\phi}\left[\sum_{t=1}^{T}\left(\beta_{pred}L_{pred}(\phi) + \beta_{dyn}L_{dyn}(\phi) + \beta_{rep}L_{rep}(\phi)\right)\right] \qquad (1)$$

$$\mathcal{L}_{pred}(\phi) = -\ln p_\phi(x_t \mid z_t, h_t) - \ln p_\phi(r_t \mid z_t, h_t) - \ln p_\phi(c_t \mid z_t, h_t) \qquad (2)$$

$$\mathcal{L}_{dyn}(\phi) = \max\left(1, \mathrm{KL}\left(\mathrm{sg}\left(q_\phi(z_t \mid h_t, x_t)\right) \mid p_\phi(z_t \mid h_t)\right)\right) \qquad (3)$$

$$\mathcal{L}_{rep}(\phi) = \max\left(1, \mathrm{KL}\left(q_\phi(z_t \mid h_t, x_t) \mid \mathrm{sg}\left(p_\phi(z_t \mid h_t)\right)\right)\right) \qquad (4)$$

在訓練的時候，因為考慮到 Predictor 跟真實世界是具有 Noise 的，因此加入了 Robust Prediction 的 loss，旨在使用 two hot 的 prediction 來避免單一的迴歸錯誤。

2.1.2 演員-評論家 Actor-Critic Network

在一般的 model-based Rl 之中，Actor-Critic 是一個非常常見的 learning 方式，並且配合不同的 learning 方法，比如使用 Bellman Optimality equation 得到的 Temporal Difference Estimation，並且遵照以下 function，使用 state 得到 action 及預測的 Return。

Actor: $a_t \sim \pi_\theta(a_t \mid s_t)$, $\max_\theta E_{(s,a)\sim\mathbb{D}}[\log \pi_\theta(a|s)]$ Critic: $v_\psi(R_t \mid s_t)$

但在 DreamerV3 中，為了鼓勵更多的探索性行為，因此使用了 lambda-return 為 0.95的 learning 方法(式7)，結合了 Monte-Carlo (式5)以及 TD Estimation (式6)，在想像展開(Imagination Rollout)過程中平衡穩定性與樣本效率(Sample Efficiency)。

$$R_t^0 = r_t + \gamma c_t v_t \qquad (5)$$

$$R_t^1 = \sum_{k\geq 0}\left(\prod_{j=0}^{k-1}\gamma c_{t+j}\right) r_{t+k} \qquad (6)$$

$$R_t^\lambda \doteq r_t + \gamma c_t\left((1-\lambda)v_t + \lambda R_{t+1}^\lambda\right) \qquad R_T^\lambda \doteq v_T \qquad (7)$$

Critic Learning: 使用式8進行學習

$$\mathcal{L}(\psi) \doteq -\sum_{t=1}^{T} \ln p_\psi\left(R_t^\lambda \mid s_t\right) \qquad (8)$$

Actor Learning:

為了避免 Actor 在做 learning 的時候，因為錯誤的 scale 需要過度的調整參數及前後學習率問題，因此加入了 Scale Factor S ，同時也為了增進稀疏獎勵(Sparse Rewards)，也增加了獎勵的 Entropy: (式9, 10)。

$$\mathcal{L}(\theta) \doteq -\sum_{t=1}^{T} \mathrm{sg}\left(\left(R_t^\lambda - v_\psi(s_t)\right) / \max(1, S)\right) \log \pi_\theta\left(a_t \mid s_t\right) + \eta\, H[\pi_\theta(a_t \mid s_t)] \qquad (9)$$

$$S \doteq \mathrm{EMA}\left(\mathrm{Per}\left(R_t^\lambda, 95\right) - \mathrm{Per}\left(R_t^\lambda, 5\right), 0.99\right) \qquad (10)$$

而在 DreamerV3之中，對於 Actor-Critic 的構型如 Fig 2-1, 2-2

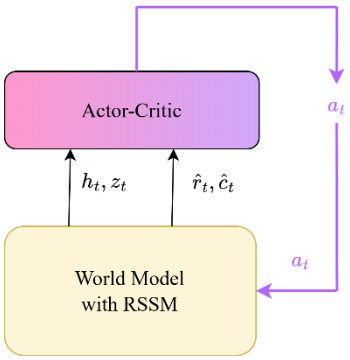


Fig. 2-1 Actor 在潛在空間中進行學習，並且在想像的軌跡(Trajectories)上使用 On model, Off-env 的學習方法

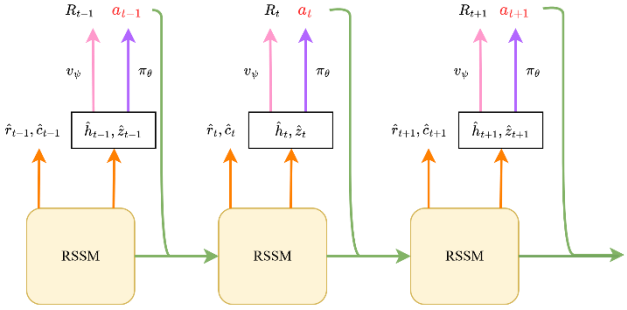


Fig. 2-2 Actor Critic 與 RSSM 互動的過程
綠色為f_ϕ 的 Sequence function
橘色為p_ϕ 的 predictive function
粉紅色為 Critic Network 預測的 Return

2.2 Advantaged Weighted Actor-Critic (AWAC)

在常見的 Actor-Critic Network 中，Actor 的 learning 方式常為:

$$\text{Actor:} \quad a_t \sim \pi_\theta(a_t \mid s_t), \qquad \max_\theta E_{(s,a)\sim\mathbb{D}}\left[\log \pi_\theta(a|s)\right] \qquad (11)$$

而在 AWAC 中，經 KL-散度、KKT condition 以及最大似然估計(MLE)，演化出:

$$\max_\theta E_{(s,a)\sim\mathbb{D}}\left[w(s,a)\log \pi_\theta(a|s)\right], \qquad w(s,a) = exp\left(\frac{max(A^\pi(s,a),0)}{\beta}\right) \qquad (12)$$

AWAC 的作法使得 Actor-Critic 可以從 On-Policy 轉變為 Off-Policy 學習，並且可以很好的轉換 Off-Policy 以及 On-Policy 之間的差距。

AWAC 也使用 offline 的 prior dataset，供 Online 以及 Offline 的經驗進行混合，抽樣學習，讓學習效率能在 Online 時避免災難性遺忘(Catastrophic Forgetting)Offline 的訓練結果，以及避免 Model Online 學不到，只會 Offline 的結果。

2.3 Information Theory: 熵(Entropy) & 散度(Divergence)

Entropy 的兩個主要作用

在強化學習及機率模型中，熵（Entropy）扮演兩個核心角色：

1. 探索的正則化項（Exploration Regularization）

在策略最適化中，提高策略分佈的熵能避免模型過早收斂至次佳策略。

高熵促使策略在動作空間中維持足夠的隨機性，從而進行更廣泛的探索。

因此，許多 RL 演算法（例如 Soft Actor-Critic）直接將熵加入目標函數，以鼓勵探索，如式9，加入了一個熵探索項。

2. 不確定性的度量（Measure of Uncertainty）

熵亦可視為模型對其動作分佈之「不確定性」的量化指標。

低熵 (Low Entropy) → 分佈尖銳 → 高自信度

高熵(High Entropy) → 分佈平坦 → 低自信度

這使熵成為比較不同策略、選擇動作來源、或設計 gating / switching 模組時的重要信號。

2.3.1 香農熵(Shannon Entropy)

對於具有 K個離散狀態的機率分佈 $p(x)$，Shannon entropy 衡量平均不確定性：

$$H(p) = -\sum_{i=1}^{K} p(x_i)\log\, p(x_i).$$

均勻分佈(uniform distribution)具有最大熵，而集中於單一類別的分佈熵為零。

2.3.2 差分熵(Differential Entropy)

類似於香農熵，差分熵也是計算 p 的平均不確定性，如下式:

$$h(p) = -\int p(x)\log\, p(x)\, dx.$$

但差分熵具有更多特點:

1. 可能是負的，

2. 對尺度變換敏感，

3. 不具參數化不變性。

儘管如此，微分熵仍是衡量連續模型（例如高斯策略）不確定性的重要方式。

2.3.3 交叉熵(Cross Entropy)

對於真實分佈 $p(x)$與模型分佈 $q(x)$，交叉熵定義為：

$$H(p,q) = -\sum_x p(x)\, log\, q\,(x) \qquad \text{(離散)}$$

$$H(p,q) = -\int p(x)\, log\, q\,(x)\, dx \qquad \text{(連續)}$$

交叉熵衡量使用基於 q的最佳編碼來描述來自 p的樣本所需的平均成本。

並且它可分解為：

$$H(p,q) = H(p) + \mathrm{KL}(p \parallel q).$$

使其成為分類與密度估計中常用的損失函數。

2.3.4 KL 散度(KL Divergence)

KL 散度衡量分佈 p與 q之間的差異：

$$\mathrm{KL}(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \qquad \text{(離散)},$$

$$\mathrm{KL}(p \parallel q) = \int p(x) log\, g \frac{p(x)}{q(x)}\, dx \qquad \text{(連續)}.$$

KL 散度具有以下性質:

- $\mathrm{KL}(p \parallel q) \geq 0$,
- Zero iff $p = q$ almost everywhere,
- $\mathrm{KL}(p \parallel q) \neq \mathrm{KL}(q \parallel p)$.

KL 散度被廣泛用於強化學習、變分推論與機率模型中。

2.3.5 對稱 KL 散度(Symmetric KL Divergence)

由於 KL 散度不對稱，許多應用中更偏好其對稱版本：

$$\mathrm{SKL}(p,q) = \frac{1}{2}[\mathrm{KL}(p \parallel q) + \mathrm{KL}(q \parallel p)].$$

對稱 KL 提供更公平的分佈差異衡量方式，尤其適用於兩者角色等價的情境（例如比較兩個策略）。

2.3.6 正規化熵(Normalized Entropy)

Entropy values depend on the number of classes (discrete) or variance scale (continuous).

To obtain an interpretable measure in [0,1], entropy can be **normalized**:

- **離散**

  最大熵 = $\log K$，正規化形式為：

$$\widehat{H} = \frac{H}{\log K} \in [0,1].$$

- **連續**

  若事先知道熵落在 $[H_{\min}, H_{\max}]$範圍內，則定義：

$$\widehat{H} = \frac{H - H_{\min}}{H_{\max} - H_{\min}} \in [0,1].$$

  正規化熵能提供跨不同動作空間、模型與分佈類型之間的統一不確定性尺度。

3. Research Methodology

3.1 Problem Formulation

在 2.1中，我們理解到了 DreamerV3使用兩個部分: World Model: RSSM 學習環境的變化； Actor-Critic 在想像中學習策略，但 DreamerV3的策略並不是經過真實經驗學習的，因此我們希望學習一個新的策略，學習真實的歷史經驗。

但這會造成兩個的策略衝突，因此需要兩個策略的選擇機制。

總而言之，遇到的兩個問題:

1. DreamerV3無法學到真實經驗，須要有另一個策略進行學習。
2. 若有另一個策略，需要對 DreamerV3以及新策略進行選擇。

3.2 Method

面對這兩個問題，我們分別提出兩個解方:

1. 使用2.2節的 AWAC 學習真實的經驗，並且使用其離線策略(Offline-Policy)以及加權的特性加強好經驗。

2. 使用資訊理論(Information Theory)的方式，使用免訓練(Training-Free)的方法進行選擇，並且這個方法可以衡量兩個策略的信心程度、差異程度進行選擇，我們稱之為閘選擇機制(Gating Mechanism)。

3.2.1 Experience-Based AWAC

我們設計了一個和 DreamerV3一樣大小的 AWAC，但是我們使用與原本的 AWAC 些許不同的學習方式。

我們使用了近似 式12作為 Actor 的 learning 方法，但是我們將獎勵/回報(reward)進行了 scale，因此我們的 Actor 學習方式如下:

$$S \doteq \mathrm{EMA}\left(\mathrm{Per}\left(R_t^{\lambda}, 95\right) - \mathrm{Per}\left(R_t^{\lambda}, 5\right), 0.99\right) \qquad (13)$$

$$\max_{\theta} E_{(s,a)\sim\mathbb{D}}\left[w(s,a)\log\pi_{\theta}(a|s)\right], \quad w(s,a) = exp\left(\frac{max(A^{\pi_{slow}}(s,a),0)}{\beta}\right) \qquad (14)$$

$$A^{\pi_{slow}}(s,a) = \frac{R_t}{S} - v_{\psi_{slow}}(R_t \mid s_t) \qquad (15)$$

而我們的 critic network 共用了和 dreamerV3 的 Critic network，並且我們使用其 slow network，避免 learning 時的 noise 影響過大。

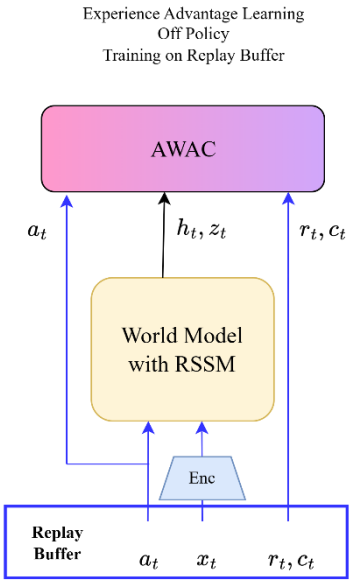


Fig. 3 AWAC 的訓練方法。
我們將真實世界的訊號傳入 AWAC，但為了讓資訊對齊，因此我們也將 z_t傳入，以讓 AWAC 的訓練是在經驗(真實)的訊號而非想像的資訊上。

此外，我們的 AWAC 會使用來自真實世界的訊號，並且這個過程也可以讓 AWAC 對齊真實經驗與想像的差距，另外我們使用的是 replay buffer，不同於原論文的設計，我們使用了 replay buffer 而不是整個 dataset，讓他能對近期的記憶更敏感，也

會對訓練時或者 lifelong learning 時的負擔更小。

3.2.2 Gating Mechanism

我們使用了一個基於熵以及 KL-散度的方法(可參考2.3節)，演算法如下:

Algorithm 1 AWAC Gate at Policy Time

Input: features x_t, policies $\pi(\cdot \mid x_t)$, $\pi^A(\cdot \mid x_t)$
Sample $a \sim \pi$, $a^A \sim \pi^A$;
for each head k **do**
  Compute $\hat{H}_k(\pi_k)$ and $\hat{H}_k(\pi_k^A)$ using bounded entropy.
  Compute $\mathcal{D}_k = \frac{1}{2}\big(\mathrm{KL}(\pi_k \| \pi_k^A) + \mathrm{KL}(\pi_k^A \| \pi_k)\big)$.
end for
$\Delta\hat{H} \leftarrow \mathrm{mean}_k\big(\hat{H}_k(\pi_k) - \hat{H}_k(\pi_k^A)\big)$; $\mathcal{D} \leftarrow \mathrm{mean}_k(\mathcal{D}_k)$.
$\mathrm{penalty} \leftarrow \tanh\big(\max(\mathcal{D} - \tau, 0)/s\big)$.
$\mathrm{score} \leftarrow \Delta\hat{H} - \beta_{div} \cdot \mathrm{penalty} - \mathrm{bias}$.
$p^*_{gate} \leftarrow \sigma\big(\alpha \cdot \mathrm{score}\big)$; $p_{gate} \leftarrow \mathrm{clip}(p_{gate}, p_{\min}, p_{\max})$.
$m \sim \mathrm{Bernoulli}(p^*_{gate})$ independently per environment.
$\mathrm{action} \leftarrow \mathrm{where}(m, a^A, a)$.
Return action and log metrics $\{p_{gate}, m, \Delta\hat{H}, \mathcal{D}\}$.

演算法 1

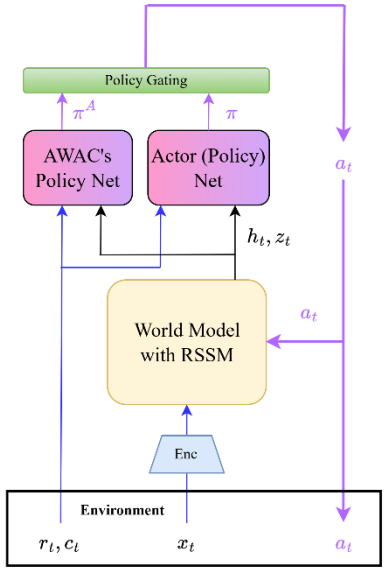


Fig. 4
我們的模型在進行推論時的架構。

分為3個步驟:

1. 計算熵以及對稱 KL 散度
2. 使用正規化的熵以及經過平均、tanh 的對稱散度計算p_{gate}
3. 使用 sigmoid 函數並用 Bernoulli 隨機抽取 policy，並且在眾多 policy 中也抽取行為(action)

我們的方法分成3個步驟，而核心的概念是:

使用熵來確認兩個 policy 對於自己決策的確信度，並且透過對稱 KL 散度計算 AWAC 以及 DreamerV3的 Actor Network 之間的決策差距，並使用正規化、平均、hyperbolic tangent、clipping 等方法將兩者變成同一規模，最後再使用 Bernoulli 隨機進行抽取行為。

正規化熵我們已經於2.3.6節提過，而對稱 KL 散度的規模，我們使用了 Fig 5的 hyperbolic tangent、clipping 完成:

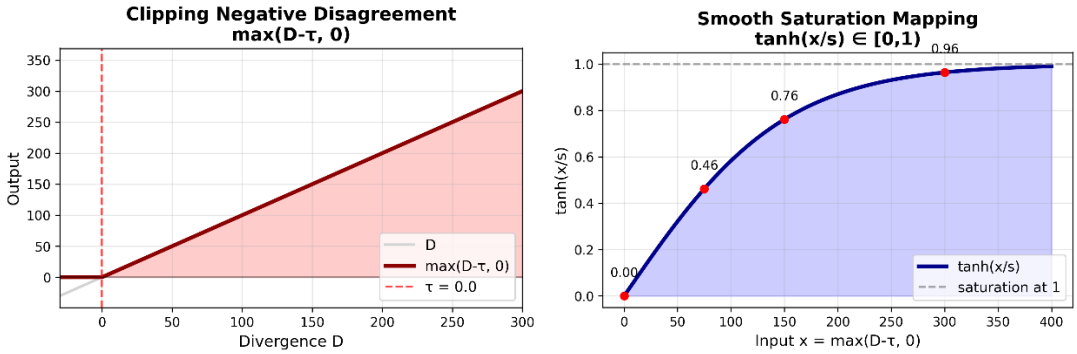


Fig. 5

使用 clipping，並且尋找到一個 saturation s 可以使得 D 不會過度突發造成選擇可能坍縮

最後使用 Sigmoid 搭配 Bernoulli 進行選擇行為(action)。

4. Experiment

我們測試了3個 task:

DeepMind Control Suite (DMC) proprio walker walk, DeepMind Control Suite (DMC) catch cup[4], Crafter [5]。驗證了在各種任務上，包含穩定走路的 walker，或者動態的杯子接球任務(catch cup)，或者開放性世界中(crafter)，都有著良好的學習效果，可超越或匹敵原 DreamerV3的方法，並且這種混合策略也確實使用了不同的策略進行選擇，可參考以下表格。 (Table 1)

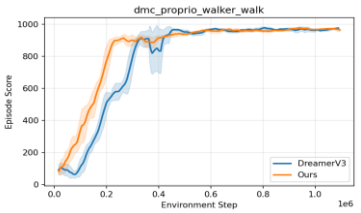
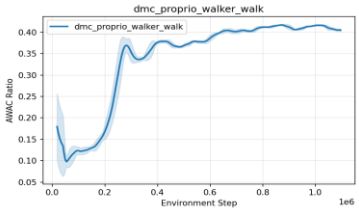
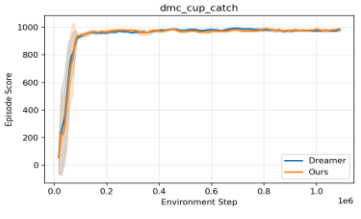
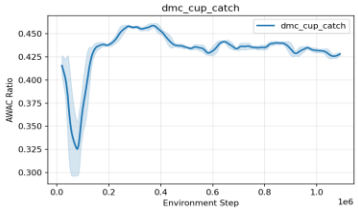
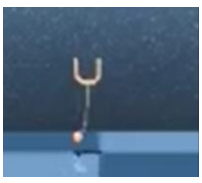
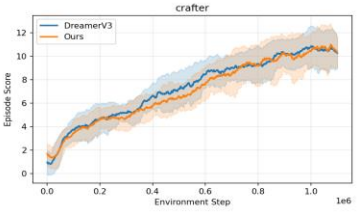
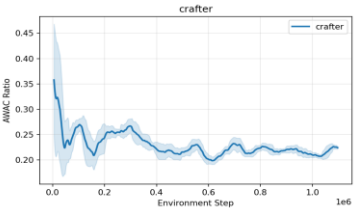
DMC Proprio Walker walk				
DMC Catch Cup				
Crafter				

Table 1 三種不同的 task 上的結果，從左至右是分數、選擇機率、遊戲畫面

5. Conclusion

我們從 World Model DreamerV3，改進其運行模式，使其能不只有想像的策略，也有基於經驗的策略；而後我們也建立了一個行為選擇機制 (Gating Mechanism)，這些基於 Information Theory 的方法也常被使用於現代的 MoE (混和專家結構，Mixture Of Experts)，以及 LLM 的多代理人(Agent)選擇時，使用以熵為主計算的置信度/信心/自信度；而 KL 散度也被用於多代理人(Agent)間決策差異度的計算方法。

最後我們也提出了一個有效的行為選擇機制，投入了使用也具有高價值，可以匹敵 DreamerV3本身的策略，而此方法也不是單純的從工程層面完成，而是基於許多的數學統計原理，包含了資訊理論等。

我們認為未來這種方式具有高度潛力，並且能更多的被推廣再未來小模型，多 Agent 的場景中。

6. Reference

[1] *Sutton, Richard S. and Andrew G. Barto. "Reinforcement Learning: An Introduction." IEEE Transactions on Neural Networks 16 (2005): 285-286.*

[2] *Hafner, Danijar, J. Pašukonis, Jimmy Ba and Timothy P. Lillicrap. "Mastering diverse control tasks through world models." Nature 640 (2025): 647 - 653.*

[3] *Nair, Ashvin, Murtaza Dalal, Abhishek Gupta and Sergey Levine. "Accelerating Online Reinforcement Learning with Offline Datasets." ArXiv abs/2006.09359 (2020): n. pag.*

[4] *Tassa, Yuval, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap and Martin A. Riedmiller. "DeepMind Control Suite." ArXiv abs/1801.00690 (2018): n. pag.*

[5] Hafner, Danijar. 2022. "Benchmarking the Spectrum of Agent Capabilities." Paper presented at the *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2109.06780

7. Review and reflections

這個專題我們認為非常難做，因為很考驗把高層次的想法實作，需要從模型本身考慮 RL 策略，是否在數學上有最佳化或者偏誤問題，也需要觀察做策略選擇，還要做基於 jax 語言:一種相較於 Pytorch 以及 tensorflow 架構再更底層的架構，我們同時要處理 CPU/GPU 之間的資料協調問題。
在暑假，我們幾乎每天都在討論要怎麼做，並且尋找資源，也在8月底時跟教授討論時被要求重做某一個部分，但是這些過程都對我們來說非常重要，因為這個過程讓我知道需要時常檢視自己的研究，跳出現在的研究狀態，看有沒有甚麼麼高層次的問題。

某一次的會議被電得很慘，也是我學到最多的一次:
1.講清楚論文要解決的事情
2.論文的方法以及他所要解決問題的關聯
3.論文是不是真的有新穎程度，或者足夠的創見
4.從高層次去想論文的合理性以及去找基線來驗證他的目的
5.有條理地去整理思維，不要只用同樣的話去陳述論文，用不一樣的話來說，因為同樣的話再講一次還是聽不懂
6. 保持心態，如果心態不好就沒辦法承受住教授的問題
7.清楚每一個細節的設計原因

最後，我想推薦各位一間好吃的餐廳: 彈芽麵，google map 資訊如下:

https://maps.app.goo.gl/fwoUMnC2rQwL9Xxp6

8. Plan Management and Teamwork

江承瀚: 程式編寫, 提案, 報告 (45%)

黃祥陞: 程式編寫, 實驗, 報告(55%)

英文版報告也可參考，按照 IEEE 格式書寫

https://github.com/faidavid7/Reinforcing-Good-Experiences-AWAC-DreamerV3-with-Confidence-Aware-Policy-Gating