Climbing the Reasoning Plateau: Integrating the Free Energy Principle and Cognitive Science for Advanced AI Models

Feng Li

July 2024

Abstract

Artificial Neural Networks (ANNs) have revolutionized fields such as computer vision, natural language processing, and speech recognition. Despite their successes, current ANN models face limitations, including hallucination, bias, and resource intensity. These challenges are attributed to the innate limits of ANNs, which often prioritize memorization over deeper reasoning. This article proposes a theoretical framework to analyze the innate limitations of ANNs in reasoning. The framework is applied to assess methods such as Synthetic Biological Intelligence(SBI), Energy-Based Models(EBM), and Active Inference. The paper aims to help those seeking to understand why LLMs struggle with reasoning and to expand their toolkit by integrating insights from cognitive science and biology within a computer science perspective.

1 Introduction

The impact of ANNs on modern life is undeniable, particularly in fields such as computer vision [1], natural language processing [2–4], and speech recognition [5]. These advancements are driven by innovations such as convolutional neural networks [6], recurrent neural networks [7–9], generative adversarial networks [10], transfer learning [11], attention mechanisms [12, 13], residual connections [14], and reinforcement learning with human feedback [3]. However, the limitations of current ANN models are becoming increasingly evident [15, 16].

Recent studies highlight significant issues in large ANN networks, including hallucinations and deficiencies in logical reasoning and coherent decision-making abilities [17–20]. These shortcomings are primarily due to the fundamental constraints of ANNs, which emphasize minimizing discrepancies between predicted and actual outcomes. As a result, these models often prioritize memorization over discovering deeper connections[21]. Although some evaluations and training tasks have been specifically designed to mitigate these issues [22], they fail to overcome the core limitations in building comprehensive world models that are essential for active reasoning and prediction.

This article proposes a theoretical framework to analyze the shortcomings of prevalent ANN models and reviews methods aimed at constructing world models, including energy-based models, Synthetic Biological Intelligence, and Active Inference. These approaches seek to overcome existing limitations and advance towards more sophisticated artificial intelligence capable of higher cognitive processes. The paper aims to provide a theoretical guide for those grappling with the reasoning plateau and those interested in leveraging insights from cognitive science to improve current methods.

In section 2, we introduce a theoretical framework with unified notation to streamline terminology and nomenclature, which will be beneficial for discussing subsequent methods. In section 3, we translate ANN models into this framework, providing a detailed explanation of the issues mentioned earlier. section 3.1 covers energy-based models, applying them within the framework to address their static nature. section 4 introduces the Free Energy Principle (FEP), explaining how minimizing free energy can lead to more effective models as a function of the following methods. In section 5, we analyze various implementation methods based on FEP. Specifically, section 5.1 discusses and analyzes biological implementations, while section 5.2 explores active inference through Partially Observable Markov Decision process (POMDP). Finally, in section 6, we summarize our findings and suggest future research directions in this promising field.

2 Unified Intelligence Framework

To formally analyze ANN models and those aimed at overcoming the reasoning plateau, it is crucial to establish a general framework that can encompass both. This is particularly important in addressing the challenges posed by inconsistent notation, terminology, and nomenclature across disciplines. A unified framework will serve as a common language, enabling rigorous comparison and analysis of these models.

Generally, the problem of intelligence can be described as an intelligent agent a operating within an environment, where the environment is composed of various states, usually referred to as hidden states or latent states, denoted by h. The environment transit from state to state in accordance with the second law of thermodynamics, which states that every system tends toward dissipation, as shown in fig. 1. In other words, each state evolves from one to another in the direction of higher entropy.

The primary objective of an agent is to interact with its environment to maintain homeostasis, thereby resisting entropic forces amidst varying states. Without this resistance, the agent would merge into the environment and lose its distinct identity (see fig. 1 (b)).

The agent possesses its own sensory system, which senses the environment [1]

[1]The environment could include parts of the agent itself that need to be sensed by the sensory system. For example, an animal would have no awareness of its limbs without actually feeling or seeing them.

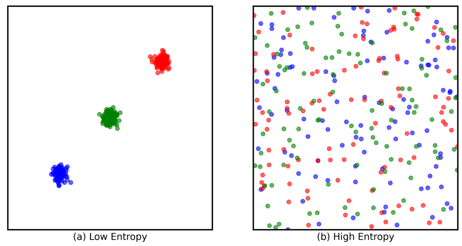

Figure 1: Entropy evaluates the level of disorder in a system. According to the second law of thermodynamics, any closed system tends to transition from a low entropy state (a) to a high entropy state (b). In (a), there are three distinct clusters of nodes, each of which can be considered an object. For each object, all others are part of the environment. As entropy increases, these objects can no longer be distinctly identified as they blend into the environment.

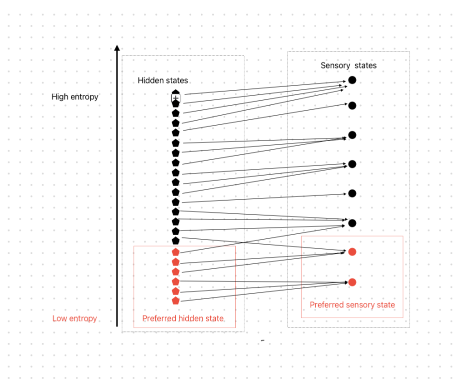

Figure 2: Mapping between Hidden States and Sensory States.

and generates sensory states, denoted s. However, the size of the sensory state set $\mathcal{S}$ is much smaller than the set of environmental states $\mathcal{H}$. Consequently, the mapping between environmental states and sensory states is not one-to-one due to the limited precision of the sensory system.

To remain within low-entropy hidden states $\mathcal{H}^\star \subset \mathcal{H}$, the agent must take actions y to influence the environment. Due to the inaccessibility of the hidden states, the agent infers these states by observing whether the sensory states falls within the set of $\mathcal{S}^\star \subset \mathcal{S}$, known as the preferred sensory state, corresponds to $\mathcal{H}^\star$, as shown it fig. 2.

To make optimal decisions that lead to preferred sensory states, the agent needs to use sensory states to infer hidden states, effectively reversing the many-to-one mapping.This process involves determining a probability density function, known as the posterior belief $p(h \mid s)$, which answers the question of which hidden state caused the given sensory state. Using on Bayes' rule, it can be computed using prior beliefs $p(h)$, the likelihood model $p(s \mid h)$, and model evidence $p(s)$.

$$p(h \mid s) = \frac{p(h)p(s \mid h)}{p(s)} \tag{1}$$

However, the marginalization operation might be analytically intractable, especially when there is a near-infinite number of h mapping to the same s in complex cases. Variational (Bayesian) inference approximates the posterior belief using $q_{m_\theta}(h)$, where m denotes the agent's inference model with its updatable model parameters θ (in this article, m_θ will be dropped for clarity).

The agent's goal is to minimize the divergence between the true posterior belief $p(h \mid s)$ and the approximate posterior belief, thereby improving its inference of hidden states. Based on these inferences, the agent evaluates possible actions y from the action set $\mathcal{Y}$ using eq. (2) to score each action, where f is the score function, and $y^\star$ is the optimal action.

$$y^\star = \underset{y \in \mathcal{Y}}{\arg\max}\, f(y, q(h), \mathcal{S}^\star) \tag{2}$$

While different actions may have specific targets, such as reducing uncertainty about hidden states, they ultimately aim to guide the agent towards the preferred hidden state $\mathcal{H}^\star$. After an action affects the environment, the hidden state transitions to a new state, and the process begins anew. At the same time, the agent also have capability to update its approximation q to better match p , leading to improved predictions of hidden states and stabilization in the environment.

In the following sections, we will analyze different approaches within this context, focusing on ANN models and those aimed at advancing beyond the reasoning plateau.

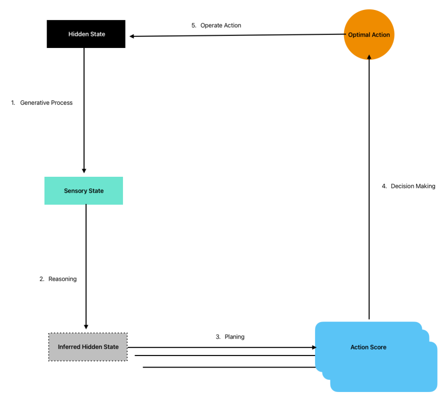

Figure 3: A cycle where the agent uses sensory data to deduce hidden states, then takes optimal actions to steer the hidden state towards a preferred direction.

3 ANN Limitations in Unified Framework

Transfer the ANN model in the framework mentioned above. The sensory state s corresponds to the data point as model input, while y corresponds to the model output for the specific data point. The aggregate of the actions' score corresponds to model logits.

The ANN does not aim to simulate the full feedback loop but instead focuses on processing sensory data to determine the optimal action, as shown in the fig. 3. Since there is no direct way to access the probability densities p and q, one uses mathematical tools such as KL divergence to minimize the divergence between the two probability densities. Instead of directly evaluating this divergence, ANN models often use a surrogate approach by introducing $\hat{y}$, the ground truth action annotated or augmented as in self-supervise learning, by the model designer.

Since both y and $\hat{y}$ are often discrete, it is straightforward to define a cost function $\mathcal{L}$, such as cross-entropy loss, to measure the difference between y and $\hat{y}$. This difference serves as the basis for optimization using gradient descent.

$$\mathcal{L} = \|\hat{y} - y^{\star}\| \tag{3}$$

ANNs reduce complexity by breaking the feedback loop, simplifying the problem to focus solely on sensory data and optimal actions, as shown in fig. 4. With the help of optimization methods such as gradient descent, this approach has significantly contributed to the success of AI today, enabling numerous downstream applications. However, it also limits the capabilities of models in several ways.

Firstly, breaking the feedback loop removes the dynamics that drive the

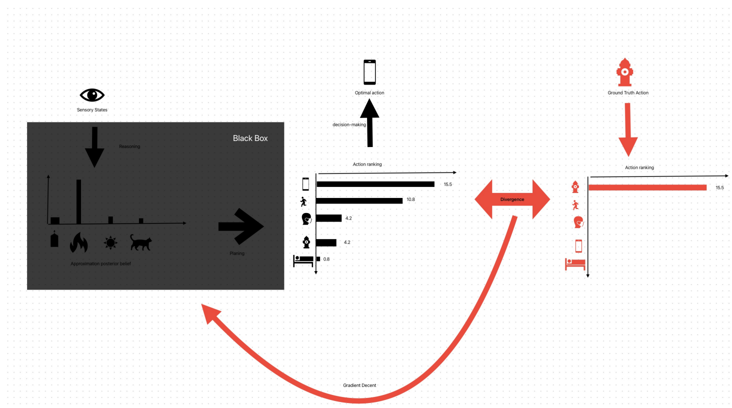

Figure 4: ANN under framework

model's approximate density q closer to the true distribution p. As a result, the model requires continuous input of shuffled data and relies on gradient descent as the primary motivation for updates, rather than allowing the agent to actively sense the environment and autonomously evaluate its actions.

By focusing solely on end-to-end approaches, parts of the agent effectively become a black box. The approximate density q, the score function, and the normalization of different actions are all obscured within this black box. This lack of transparency makes it difficult to ensure that these components are meaningfully represented within the model, raising concerns that the network might simply be functioning as a large hashtable. Some studies suggest that LLMs struggle with tasks requiring reasoning, particularly in areas like mathematical deduction. The opacity of the black box complicates efforts to control the model through prompting, and the potential reliance on a large hashtable necessitates vast amounts of training data. This approach also demands increasing numbers of parameters and consumes significant energy for task performance [23], in stark contrast to the human brain[24].

Since the ground truth actions are annotated and generated by humans, the agent's performance is heavily dependent on the quantity and quality of these ground truth actions. This reliance inherently limits the model's potential, confining ANNs to simplifying tedious tasks rather than engaging in creative work. Even in AI-generated content tasks, the model might merely compute an average of the training data based on the given prompt.

These limitations underscore the need for more advanced frameworks that can enhance reasoning abilities and reduce dependence on human-generated ground truth actions for improved performance.

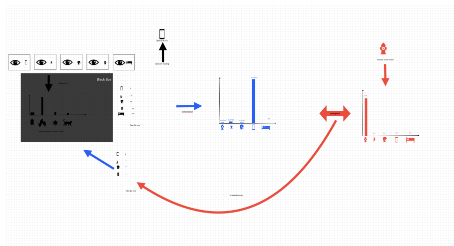

Figure 5: Energy Base model under unified framewok

3.1 Energy-Based Models

Energy-Based Models (EBMs) were first introduced by [25] in 2003 to generalize independent component analysis. Later, [26] expanded on this concept to unify various machine learning domains within a single framework, leveraging the concept of energy. EBMs are distinct in their use of energy to quantify the compatibility between an input s and an output $y^{\star}$. Unlike ANNs, which explicitly output probabilities or logits, EBMs produce a non-negative scalar energy value G (using the same notation as Expected Free Energy in section 5.2) for each possible input during inference through an energy function E_{θ} (where we drop the parameter θ for consistency with earlier equations). Lower energy values indicate a higher preference by the model, while higher energy values correspond to lower preference.

Formally, the energy function can be described as follows:

$$G = E(s) \tag{4}$$

$$p(y|s) = \frac{\exp\left(E(s)\right)}{\sum_{s \in \mathcal{S}} \exp\left(E(s)\right)} = \frac{\exp\left(E(s)\right)}{Z} \tag{5}$$

Since EBMs primarily focus on the energy score of single action, obtaining a probability distribution as required in generative tasks necessitates computing the partition function Z as shown in eq. (5). This involves evaluating all possible actions and explicitly normalizing them fig. 5. In contrast, ANN models generate logits that implicitly consider all possible s, allowing for direct normalization.

The goal of training EBMs is to learn an energy function that reflects the system's natural preferences. However, computing the partition function is computationally challenging, making it difficult to train EBMs using gradient descent methods, as each update requires this computation. Although some methods, such as Markov Chain Monte Carlo (MCMC), can approximate the partition function, they do not fully alleviate the computational burden [27]. Consequently, EBMs are commonly used in tasks involving comparisons, where the partition function cancels out, eliminating the need for integration and summing.

In summary, EBMs expose the limitations of the ANN framework concerning the range of problems it can solve. While ANNs rely on implicit normalization, requiring a normalized dataset, researchers have invested significant effort in constructing datasets and developing advanced normalization techniques. EBMs, by contrast, offer greater flexibility by removing the need for normalization, theoretically enabling them to solve a broader range of problems. However, due to the need for probability density estimation during inference and optimization, EBMs face the intractable partition function problem, which is only partially mitigated by approximate methods.

When integrating the EBM approach into a unified framework, as shown in fig. 5, it shares similarities with ANN models in that it only handles the process from sensory data to optimal action and cannot function as an autonomous learning model. While EBMs reveal more about the planning step and define the energy function to give the model a world representation, expanding the task domain, they also introduce the intractable partition function. Additionally, like ANN models, EBMs require ground truth feedback, sharing the same limitations, such as a lack of reasoning ability and dependence on annotated data.

4 The Free Energy Principle

This section delves into the theoretical aspects of FEP without delving into implementation details, as it forms the foundational basis for the methodologies discussed in the subsequent sections.

FEP, proposed by Karl Friston, unifies several global brain theories that explain the optimization behavior of agents, including the Bayesian brain, efficient coding hypothesis, cell assembly & correlation theory, and Neural Darwinism. It introduces the concept of free energy as a common objective of perception and action, which can be formulated as the minimization of the discrepancy between the model and the world, a notion not traditionally noted in cognitive science.

The free energy can be expressed in three forms as shown in eqs. (6) to (8):

$$F = \underbrace{\mathbb{E}_{q(h)}\left[\ln \frac{q(h)}{p(h \mid s)}\right]}_{\textit{Divergence}} \underbrace{- \ln p(s)}_{\textit{Surprise}} \tag{6}$$

$$= \underbrace{D_{KL}[q(h) \mid\mid p(h)]}_{\textit{Complexity}} - \underbrace{\mathbb{E}_{q(h)}[\ln p(s \mid h)]}_{\textit{Accuracy}} \tag{7}$$

$$= \underbrace{-\mathbb{E}_{q(h)}[\ln p(s, h)]}_{\textit{Energy}} - \underbrace{\mathbb{H}[q(h)]}_{\textit{Entropy}} \tag{8}$$

In eq. (6), the first term, divergence, evaluates the difference between the approximate posterior belief and the true posterior belief, while the second term quantifies the surprise of the data. Minimizing the free energy reduces this divergence and the surprise, implying that the agent's internal model of the world closely matches the real world.

In eq. (7), the first term, complexity, assesses the difference between the prior belief (the agent's recognition of the world without any sensory input) and the approximate posterior belief. This represents how much the belief needs to change (i.e., update the model) to explain the sensory data. The second term measures the accuracy of these predictions, which, with the negative sign, can be interpreted as prediction error. Minimizing the free energy finds a balance, using the simplest model that yields the least loss, akin to Occam's Razor.

In eq. (8), the first term, energy, derives from statistical physics, describing the energy required to move the system into this configuration from a baseline configuration. The second term is the entropy of the approximate posterior belief. Minimizing free energy requires the agent to find a baseline configuration that minimizes the effect to encompass all configurations and maintain high uncertainty about the hidden state without sensory inputs, following Jaynes's maximum entropy principle.

In the next section, we will review some implementations related to the Free Energy Principle, providing practical insights into how this theoretical framework is applied.

5 Free Energy Principle's Implementations

5.1 Synthetic Biological Intelligence

Synthetic Biological Intelligence (SBI) systems [28] can be broadly defined as the intentional synthesis of biological and silicon substrates in-vitro, designed to exhibit goal-directed or otherwise intelligent behavior. These systems typically do not involve whole organisms but utilize stem cell-derived neural tissues. Early research [29] has demonstrated that in vitro neurons can adaptively respond to incoming stimulation and engage in behaviors consistent with phenomena such as blind-source separation.

The use of closed-loop paradigms in in vitro neuron experiments—whereby neural activity is measured, applied to an environment, and the updated environmental information is fed back into the neural system—has become a critical area of study in SBI. A recent experiment provides significant support for FEP [30]. In this experiment, neural cultures, grown from human stem cells, were used to play a simple game called Pong. The neural agent perceives the motion of the ball through electrical stimulation and controls a paddle to move left and right to intercept and bounce the ball. Successful interceptions result in a predictable stimulus delivered across all electrodes simultaneously at 100Hz for 100ms, while failures lead to an unpredictable stimulus (150mV voltage at 5Hz for 4 seconds).

Over time, the neural cultures learn and improve their performance, as indicated by the increasing average duration of the rally. Interestingly, the agent

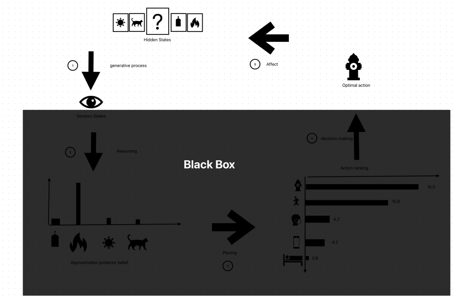

Figure 6: SBI implementation under unified framework

does not have explicit knowledge of the relationship between the stimulus type (predictable or unpredictable) and the correct action. Nevertheless, the agent tends to refine its behavior towards achieving more predictable outcomes, which corresponds to minimizing free energy.

This experiment can be framed within the unified framework. In this context, the simulated game environment represents the hidden states h, the sensory state s is provided to the cell cluster through a sensory region of the high-density multielectrode array chip, and the actions produced $y^\star$ by the neural cultures influence the hidden states by affecting the motor region of the chip.

Although this implementation demonstrates the full cycle described in the unified framework fig. 6, effectively creating an autonomous agent in a mini-game world, several limitations remain. First, a significant "black box" problem persists as the internal processes within the cell clusters are not fully understood, necessitating further advancements in neuroscience to elucidate the mechanisms at play. Second, the implementation requires sustaining the neural cultures with appropriate nutrients, which poses significant challenges for scalability with current technology.

5.2 Active Inference

Active inference, derived from the FEP, posits that agents aim to minimize Variational Free Energy to ensure that they observe states they prefer with high probability [31]. Expanding on this, active inference introduces the concept of Expected Free Energy to encompass action selection, planning, and learning. Later work by the author unified these two forms of energy as the same optimal goal for agents [32].

Under this theory, rather than passively observing data, agents continuous actively infer what future data will be observed under avaliable action[2]. They evaluate actions based on their preferred hidden states, which correspond to the types of data they wish to observe but cannot directly access. To evaluate

[2]The term "action" here is a simplification of the term *policy*, which refers to a sequence of actions defined in Active Inference, and correpdoing define in the framework for Active Inference's *action*

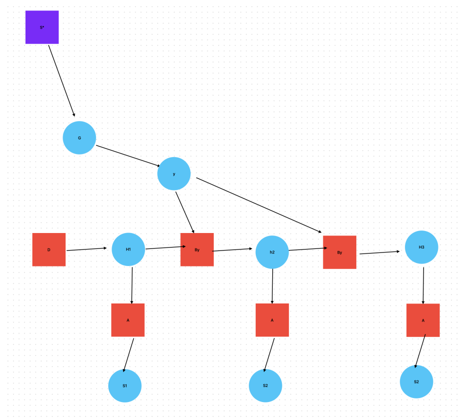

Figure 7: Partially Observable Markov Decision Process of Active Inference

different actions, agents use Expected Free Energy, which can be expressed in various forms as shown in section 2:

$$G^y = \underbrace{-\mathbb{E}_{q(s,h|y)}\left[\ln q(h|s,y) - \ln q(h|y)\right]}_{\text{Epistemic Value}} - \underbrace{\mathbb{E}_{q(s|y)}\left[\ln p(s|\mathcal{S}^\star)\right]}_{\text{Pragmatic Value}} \tag{9}$$

$$= \underbrace{D_{KL}[q(s|y)\|p(s|\mathcal{S}^\star)]}_{\text{Risk}} + \underbrace{\mathbb{E}_{q(h|y)}[H[p(s|h)]]}_{\text{Ambiguity}} \tag{10}$$

Each Expected Free Energy is computed for a specific action, denoted as G^y, where $\mathcal{S}^\star$ represents the set of observations the agent prefers to see, a subset of all possible observations $\mathcal{S}$.

In eq. (9), the first term, epistemic value, is negative information gain. By minimizing G^y, this term is maximized. The second term, pragmatic value, is the negative probability of preferred future observations. Minimizing G^y maximizes this term as well. The optimal action is the one that strikes the best balance between resolving uncertainty and moving toward the preferred future state.

In eq. (10), the first term, risk, borrowed from economics, is the divergence between predicted and preferred observed data. The second term, ambiguity, represents the level of uncertainty in the expected state.

On the implementation side, each action can be represented using a Partially Observable Markov Decision Process (POMDP) with Variational Message Passing, as shown in fig. 7.POMDP is a variant of the Hidden Markov Model (HMM). The key difference between a POMDP and an HMM is that in a POMDP, the agent has control over state transitions through its actions, whereas in an HMM, the transitions are not influenced by the agent.

The four parameters of the POMDP are as follows: D is the prior belief, representing a blind guess about the world without sensory input, stored within the agent's model. A is the Emission Probability, representing the likelihood of observations given a hidden state, also stored within the model. B_y is the Transition Probability, varying across different POMDPs for different actions y. $\mathcal{S}^\star$ is the preferred state, defined by the model designer or, in biological terms, by evolution, which is used to determine Expected Free Energy G, as shown in fig. 7.

Applying the unified framework to Active Inference, as shown in fig. 8, the real hidden state at time step n, h_n, causes a sensory state, s_n, through a generative process. The agent, besides inferring what is currently happening in the real world, h'_n, will also infer the expected hidden state at the next time step, h'_{n+1}, assuming the agent execute the action y_n.

The agent will further use h'_{n+1} to predict what sensory states s'_{n+1} will be expected, by using the stored emission probability. Accurate prediction of future events and their subsequent evaluation relies on the presence of a robust world model; without such a model, the agent lacks the necessary framework to make informed inferences and decisions.

After obtaining the s'_{n+1}, the agent could use the Expected Free Energy mentioned above to evaluate whether the s'_{n+1} is preferred by the agent. Parallel evaluations for different actions can be performed, and the action yielding the lowest G^y_{n+1} is selected as the optimal action $y^\star_n$. This $y^\star_n$ might or might not be the same as the one determined in the previous timestep. For example, if the action at timestep $n-1$ was "go eat that apple in front," the agent may start walking toward the "apple." However, at the following timestep n, as the agent gets closer to the "apple" with new sensory states, the inferred hidden state at n may reveal that the object in front is actually a box with an "apple" printed on it. The optimal action after evaluating the Expected Free Energy at timestep n could then be "open the box," thereby aborting the previous action without finishing it.

The optimal action y_n will then affect the real world by transitioning the hidden state from h_n to h_{n+1}, as the current timestep move from n to $n+1$. From agent's perspective the new hidden statesh_{n+1} will in turn causes a new round of sensory data s_{n+1} and an inferred hidden state h'_{n+1}, by the same process described for timestep n. The expected hidden state for the following timestep h'_{n+2} will be computed, but more importantly, the difference between the inferred hidden states and the expected hidden state at timestep $n+1$ will be evaluated and propagated back to the previous step to correct or support previous inferred hidden states (short-term or long-term beliefs, even memory) or adjust the emission probability and transition probability.

Active inference offers several advantages. One of the main benefits is that there is no clear distinction between training and optimization; learning occurs continuously as the agent interacts with its environment, avoiding the energy-consuming pre-train fine-tune stage before use. Additionally, this model is much closer to true intelligence, as it involves active thinking about the world rather than merely reacting to stimuli. Constructing the feedback loop can motivate the automated agent to act, infer, and learn toward the preferred sensory states.

However, there are also some limitations. Active inference models tend to be

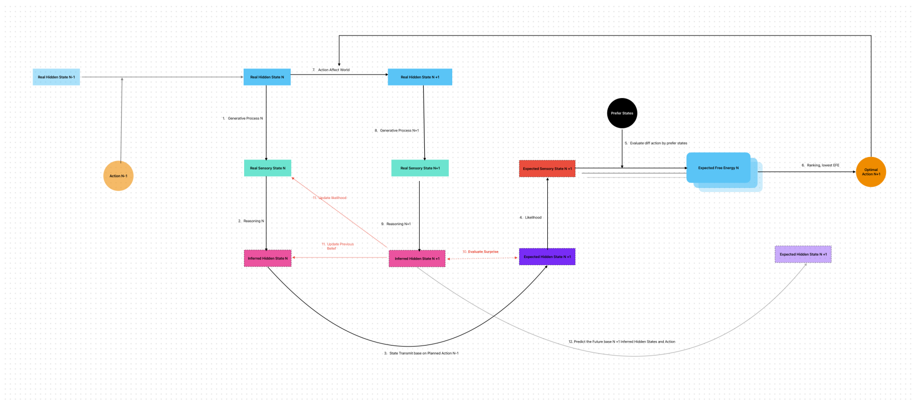

Figure 8: Active inference under the unified framework

much slower than ANNs since they require both forward and backward passes during inference, whereas ANNs typically only require a forward pass. The parameters such as emission probability, transition probability and likelihood, could be limiting as for certain tasks the potential state space is very large, making it difficult to represent using these metrics. Moreover, most applications and implementations of active inference are currently done using MATLAB, which is not open-source and can be a barrier to widespread adoption, although some packages are starting to gain popularity in Python[33]. Another challenge is the ongoing research required to accurately define the preferred states for the agent, as defining complex preferred states remains unclear. Furthermore, the model does not clearly explain how to select potential actions from the action space to compute G. If the action space is large, iterating over to find the best action will be very energy-consuming.

6 Potential Implications and Conclusion

In this article, we have explored the limitations of current ANN models and their impact on reasoning capabilities. By introducing a unified theoretical framework, we provided a consistent approach to analyzing various models, including ANNs, EBMs, and Active Inference. Our analysis revealed that while ANNs have significantly advanced AI through end-to-end optimization, they suffer from inherent limitations such as a lack of incentive to build an internal world model, reliance on human-annotated data, lack of transparency, and energy inefficiency. Although EBMs offer a more structured approach with the inclusion of a world model, they introduce challenges such as the intractable partition function and share many of the same limitations as ANNs due to their dependence on annotated data.

By integrating concepts from Active Inference, particularly through the use of POMDPs, we demonstrated how agents could potentially overcome some of these limitations by actively engaging with their environment, continuously refining their world models, and improving decision-making processes. However, the complexity of implementing these approaches remains a significant challenge, highlighting the need for further attention from the computer science field to aid in their development.

As AI continues to struggle with overcoming the reasoning plateau, drawing inspiration from cognitive science and biological processes may be crucial for advancing artificial intelligence toward more autonomous and capable systems.

References

[1] Aditya Ramesh et al. "Hierarchical text-conditional image generation with clip latents". In: *arXiv preprint arXiv:2204.06125* 1.2 (2022), p. 3.

[2] Tom B. Brown et al. *Language Models are Few-Shot Learners.* 2020. eprint: `arXiv:2005.14165`.

[3] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems.* Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 27730–27744. URL: `https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf`.

[4] Josh Achiam et al. "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774* (2023).

[5] Alec Radford et al. "Robust speech recognition via large-scale weak supervision". In: *International conference on machine learning.* PMLR. 2023, pp. 28492–28518.

[6] Yann LeCun et al. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE.* Vol. 86. 11. 1998, pp. 2278–2324. URL: `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.7665`.

[7] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "(1986) D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol. I, D. E. Rumelhart and J. L. McClelland (Eds.) Cambridge, MA: MIT Press, pp. 318-362". In: *Neurocomputing, Volume 1: Foundations of Research.* The MIT Press, Apr. 1988. ISBN: 9780262267137. DOI: `10.7551/mitpress/4943.003.0128`. eprint: `https://direct.mit.edu/book/chapter-pdf/2299556/c018389\_9780262267137.pdf`. URL: `https://doi.org/10.7551/mitpress/4943.003.0128`.

[8] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[9] Kyunghyun Cho et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.* 2014. eprint: arXiv:1406.1078.

[10] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[11] Alec Radford et al. "Improving language understanding by generative pre-training". In: (2018).

[12] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *CoRR* abs/1409.0473 (2014). URL: https://api.semanticscholar.org/CorpusID:11212020.

[13] Ashish Vaswani et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

[14] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR '16. Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90. URL: http://ieeexplore.ieee.org/document/7780459.

[15] Emily M. Bender and Alexander Koller. "Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 5185–5198. DOI: 10.18653/v1/2020.acl-main.463. URL: https://aclanthology.org/2020.acl-main.463.

[16] Emily M. Bender et al. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 610–623. ISBN: 9781450383097. DOI: 10.1145/3442188.3445922. URL: https://doi.org/10.1145/3442188.3445922.

[17] Ziwei Ji et al. "Survey of Hallucination in Natural Language Generation". In: *ACM Comput. Surv.* 55.12 (Mar. 2023). ISSN: 0360-0300. DOI: 10.1145/3571730. URL: https://doi.org/10.1145/3571730.

[18] Payal Dhar. "The carbon impact of artificial intelligence." In: *Nat. Mach. Intell.* 2.8 (2020), pp. 423–425.

[19] Isabel O Gallegos et al. "Bias and fairness in large language models: A survey". In: *Computational Linguistics* (2024), pp. 1–79.

[20] Janice Ahn et al. "Large language models for mathematical reasoning: Progresses and challenges". In: *arXiv preprint arXiv:2402.00157* (2024).

[21] Kushal Tirumala et al. “Memorization without overfitting: Analyzing the training dynamics of large language models”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 38274–38290.

[22] Xiao Liu et al. “Agentbench: Evaluating llms as agents”. In: *arXiv preprint arXiv:2308.03688* (2023).

[23] Lasse F Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. “Carbontracker: Tracking and predicting the carbon footprint of training deep learning models”. In: *arXiv preprint arXiv:2007.03051* (2020).

[24] Jason K Eshraghian et al. “Training spiking neural networks using lessons from deep learning”. In: *Proceedings of the IEEE* (2023).

[25] Yee Whye Teh et al. “Energy-based models for sparse overcomplete representations”. In: *Journal of Machine Learning Research* 4.Dec (2003), pp. 1235–1260.

[26] Yann LeCun et al. “A tutorial on energy-based learning”. In: *Predicting structured data* 1.0 (2006).

[27] Yang Song and Diederik P Kingma. “How to train your energy-based models”. In: *arXiv preprint arXiv:2101.03288* (2021).

[28] Brett J Kagan et al. “The technology, opportunities and challenges of synthetic biological intelligence”. In: *Biotechnology advances* (2023), p. 108233.

[29] Takuya Isomura, Kiyoshi Kotani, and Yasuhiko Jimbo. “Cultured cortical neurons can perform blind source separation according to the free-energy principle”. In: *PLoS computational biology* 11.12 (2015), e1004643.

[30] Brett J Kagan et al. “In vitro neurons learn and exhibit sentience when embodied in a simulated game-world”. In: *Neuron* 110.23 (2022), pp. 3952–3969.

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

[32] Thomas Parr and Karl J Friston. “Generalised free energy and active inference”. In: *Biological cybernetics* 113.5 (2019), pp. 495–513.

[33] Conor Heins et al. “pymdp: A Python library for active inference in discrete state spaces”. In: *arXiv preprint arXiv:2201.03904* (2022).