Emotions as Transferable Policy Primitives for General Intelligence

Ryuku Akahoshi
Independent Researcher
r.l.akahoshi@gmail.com

January 14, 2026

Abstract

We propose a novel neural architecture that leverages discrete emotional tokens as transferable policy primitives for general intelligence. Building upon the theoretical framework that emotions solve credit assignment in continuous action spaces, we present a practical implementation where emotional states serve as an interface between task-agnostic strategic reasoning and task-specific behavioral execution. Our architecture separates learning into two components: (1) a universal emotion-selection policy trained via Monte Carlo Tree Search that transfers across domains, and (2) task-specific reflex actors that map emotion-state pairs to concrete actions. This decomposition enables efficient transfer learning—strategic reasoning learned in one domain (e.g., Minecraft) directly applies to novel domains (e.g., FPS games) with only the reflex layer requiring retraining. We formalize this architecture mathematically and demonstrate why this separation is both computationally necessary and sufficient for achieving general intelligence in continuous control tasks.

1 Introduction

1.1 The Transfer Learning Problem

Current reinforcement learning systems face a fundamental limitation: knowledge learned in one domain rarely transfers to another. An agent mastering Minecraft cannot apply its strategic insights to first-person shooters, despite both requiring similar high-level reasoning about risk, exploration, and resource management.

This failure stems from conflating two distinct types of knowledge:

- **Strategic knowledge**: When to be aggressive vs. cautious (universal)
- **Technical knowledge**: How to aim, build, or move (domain-specific)

Traditional RL architectures learn a monolithic policy $\pi(a|s)$ that entangles both types, making transfer impossible.

1.2 Theoretical Motivation

Our approach is motivated by a key insight: emotions can serve as discrete tokens that make strategic reasoning tractable in continuous action spaces. Rather than planning over infinite possible actions ("move 5.2cm at 37.4 degrees"), we plan over a finite set of strategic stances ("be aggressive", "be cautious"). This reduces Monte Carlo Tree Search from operating over infinite continuous actions to a finite discrete set.

The separation enables transfer: strategic reasoning ("when to be aggressive") is universal across tasks, while execution ("how to be aggressive in this specific environment") is task-specific. Even with only 8 emotional tokens, the combination of rich environmental context z and strategic stance e produces diverse concrete behaviors.

1.3 Our Contribution

We propose a three-module architecture that operationalizes emotional tokens as the interface between universal strategy and domain-specific execution:

1. **Encoder**: Compresses high-dimensional observations into compact latent states
2. **Emotion-MCTS**: Selects discrete emotional tokens via tree search (transferable)
3. **Reflex Actor**: Maps (latent state, emotion) $\rightarrow$ concrete actions (non-transferable)

Key insight: When learning a new task, only the Reflex Actor requires retraining. The Emotion-MCTS module—encoding millions of iterations of strategic reasoning—transfers directly.

2 Mathematical Framework

2.1 The Hierarchical Decomposition

Let $\mathcal{S}$ denote the observation space and $\mathcal{A} \subset \mathbb{R}^d$ the continuous action space. Define:
Emotional Token Set: $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ where $|\mathcal{E}| = n \ll \infty$
The traditional policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ is decomposed as:

- **Universal Emotion Policy**: $\mu : \mathcal{Z} \rightarrow \mathcal{E}$
- **Task-Specific Reflex Policies**: $\pi_e : \mathcal{Z} \rightarrow \mathcal{A}$, $\forall e \in \mathcal{E}$

where $\mathcal{Z} \subset \mathbb{R}^k$ is a compressed latent state space with $k \ll \dim(\mathcal{S})$.
Encoder function: $\phi : \mathcal{S} \rightarrow \mathcal{Z}$
The complete system operates as:

$$s \xrightarrow{\phi(s)} z \xrightarrow{\mu(z)} e \xrightarrow{\pi_e(z)} a$$

2.2 Information Theoretic Perspective

The encoder ϕ performs lossy compression, preserving only strategically relevant information:

$$I(Z; R) \approx I(S; R) \tag{1}$$
$$H(Z) \ll H(S) \tag{2}$$

where R denotes future returns. The latent state z captures "what matters for strategy" while discarding irrelevant details.

Crucial property: The emotional policy μ operates on z, not s. This enables transfer—if two domains share similar strategic structures, their latent spaces align, allowing μ to generalize.

2.3 Why Emotions Must Be Discrete

Theorem 1 (Necessity of Discretization)**.** *For any visit-count-based learning algorithm in continuous action spaces, convergence to optimal policy is impossible.*

Proof sketch: In continuous $\mathcal{A}$, $P(a_i = a_j) = 0$ for distinct samples. Therefore $N(s, a) = 1$ almost surely, preventing statistical aggregation required for learning. □

Theorem 2 (Sufficiency of Emotional Discretization)**.** *A hierarchical policy with $|\mathcal{E}| = n$ discrete high-level actions enables MCTS with complexity $O(n^h)$ where h is planning horizon.*

2.4 The Actor's Deterministic Mapping

A critical design choice: the reflex actor $\pi_e(z)$ is deterministic for fixed (z, e).

Why this works:

- z contains rich situational information (time of day, resources, health, ...)
- e specifies strategic stance (aggressive, cautious, exploratory, ...)
- Together, (z, e) sufficiently determines optimal action

Example in Minecraft:

$z_1 = [\text{night}, \text{low\_wood}, \text{full\_health}, \ldots] + e = \text{“optimistic”} \rightarrow a_1 = \text{“chop trees outside”}$
$z_2 = [\text{night}, \text{high\_iron}, \text{full\_health}, \ldots] + e = \text{“optimistic”} \rightarrow a_2 = \text{“prepare nether portal”}$

The same emotion e produces different actions because z differs. No stochasticity needed.

3 Architecture Design

3.1 Module 1: Encoder (Reality Compressor)

Input: Raw sensory data (pixels, audio, game state)

Output: Latent vector $z \in \mathbb{R}^k$ (e.g., $k = 64$)

Training objective: Minimize reconstruction loss while maximizing mutual information with future rewards:

$$\mathcal{L}_{\text{encoder}} = \|s - \text{decoder}(z)\|^2 - \lambda \cdot I(z; R_{t:t+H})$$

The encoder learns to discard irrelevant details (exact pixel colors) while preserving strategic signals (resource levels, threat proximity).

3.2 Module 2: World Model + Value Network

Components:

- Predictor $P : (\mathcal{Z}, \mathcal{E}) \rightarrow \mathcal{Z}$
- Value function $V : \mathcal{Z} \rightarrow \mathbb{R}$

Role in MCTS: When simulating “what if I choose emotion e?”:

1. P predicts future latent state $z' = P(z, e)$

2. V evaluates that future $V(z')$

3. MCTS uses these to select optimal emotional token

Training:

- Predictor: Minimize $\|z_{t+1} - P(z_t, e_t)\|^2$
- Value: Minimize $(V(z_t) - G_t)^2$ where G_t is actual return

Key insight: The world model predicts abstract changes in strategic situation, not pixel-level details. "Optimism leads to advantage" rather than "optimism leads to specific x, y, z coordinates."

3.3 Module 3: Reflex Actor (Muscle Memory)

Input: Current latent state z_t + selected emotion e_t
Output: Concrete action $a_t \in \mathcal{A}$
Architecture:

$$a = \text{Actor}(z, e) = \text{MLP}([z; \text{one\_hot}(e)])$$

Training: Standard policy gradient or actor-critic methods:

$$\nabla J = \mathbb{E}[\nabla \log \pi_e(a|z) \cdot A(z, a, e)]$$

where A is the advantage function.
Critical property: This is the ONLY module that must be retrained for new domains.

4 The Emotion-MCTS Algorithm

4.1 Search Process

At each timestep t:

1. **Observe**: Get current latent state $z_t = \phi(s_t)$
2. **MCTS Simulation** (N iterations):
   - Selection: Traverse tree using UCB1 over emotional tokens
   - Expansion: Add new emotional branch if needed
   - Simulation: Use world model $P(z, e)$ to predict future states
   - Backpropagation: Update $Q(z, e)$ with $V(z')$ estimates
3. **Decision**: Select emotion with highest visit count

$$e^* = \arg\max_{e \in \mathcal{E}} \left[Q(z, e) + c\sqrt{\frac{\ln N(z)}{N(z, e)}} \right]$$

$$e_t = \arg\max_{e \in \mathcal{E}} N(z_t, e)$$

4. **Execution**: Reflex actor executes

$$a_t = \pi_{e_t}(z_t)$$

4.2 Why This Enables Transfer

The MCTS statistics $\{N(z, e), Q(z, e)\}$ encode strategic knowledge:

- "In resource-scarce situations, optimism rarely pays off"
- "When ahead, conservative consolidation is usually best"
- "Exploration pays off early, exploitation pays off late"

These patterns are domain-invariant. The MCTS policy μ learned in Minecraft directly applies to FPS games because both involve similar strategic trade-offs.

What changes between domains:

- The encoder ϕ (different sensory inputs)
- The reflex actor π_e (different action spaces)
- The world model P (different transition dynamics)

What stays constant:

- The emotional token set $\mathcal{E}$ (aggressive, cautious, exploratory, ...)
- The MCTS policy μ (when to be aggressive vs. cautious)

5 Emotional Token Design

5.1 Example Emotional Token Set

We propose an example set of emotional tokens that span key strategic dimensions:

Emotion	**Strategic Meaning**	**Risk/Reward Bias**
Optimistic	Maximize expected value	High risk, high reward
Cautious	Minimize variance/risk	Low risk, low reward
Exploratory	Maximize information gain	High uncertainty tolerance
Exploitative	Maximize immediate reward	Low uncertainty tolerance
Persistent	Continue current subgoal	High action inertia
Adaptive	Switch to new subgoal	Low action inertia
Aggressive	Increase action magnitude	High intensity
Conservative	Decrease action magnitude	Low intensity

5.2 Design Considerations

The optimal number and composition of emotional tokens remains an open empirical question. Several factors suggest different cardinalities:

Smaller sets ($|\mathcal{E}| \approx$ 8-16):

- Simpler MCTS (lower branching factor)
- Faster convergence of visit statistics
- Risk of insufficient expressivity

Larger sets ($|\mathcal{E}| \approx$ 50-200):

- Richer strategic vocabulary
- Better alignment with human emotional repertoire
- Higher computational cost per decision

Comparison to human emotions: Humans possess approximately 100–200 distinguishable emotional states. Whether artificial systems require similar granularity, or whether a smaller set suffices due to the richness of the latent state z, remains to be determined experimentally.

5.3 The Actor Provides Fine-Grained Control

Crucially, we don't need "optimism about exploration" vs. "optimism about combat" as separate tokens. The actor $\pi_e(z)$ differentiates:

$$z_1 = [\text{enemy\_visible}, \text{low\_ammo}, \ldots] + \text{“optimistic”} \rightarrow \text{“charge forward shooting”}$$
$$z_2 = [\text{no\_enemies}, \text{unexplored\_area}, \ldots] + \text{“optimistic”} \rightarrow \text{“explore distant region”}$$

The latent state z provides context; the emotion e provides strategic stance. Together, they uniquely determine behavior.

6 Credit Assignment and Learning

6.1 Three Types of Errors

When an episode fails (or succeeds), credit must be assigned to three distinct components:

Component	**What Failed**	**Learning Update**
Emotion-MCTS (μ)	"Choosing optimism was strategically wrong"	Update $Q(z, e)$, adjust selection probabilities
World Model (P, V)	"Predicted optimism would work, but was wrong"	Update P and V to better predict outcomes
Reflex Actor (π_e)	"Optimism was right, but execution was poor"	Update π_e to better implement emotional intent

6.2 Formal Learning Objectives

Emotion Policy (MCTS):

- Learns implicitly through tree search statistics
- No gradient updates—pure search-based optimization
- Transfers directly to new domains

World Model:

$$\mathcal{L}_P = \mathbb{E}[\|z_{t+1} - P(z_t, e_t)\|^2] \tag{3}$$
$$\mathcal{L}_V = \mathbb{E}[(V(z_t) - G_t)^2] \tag{4}$$

Reflex Actor:

$$\mathcal{L}_\pi = -\mathbb{E}[\log \pi_e(a_t|z_t) \cdot A_t]$$

6.3 Why Separation Enables Transfer

In monolithic policies, failure is ambiguous:

- "Did I choose the wrong strategy, or execute poorly?"
- "Was my prediction wrong, or my action selection?"

Our architecture disentangles these:

- Strategy (μ) learns slowly, transfers broadly
- Execution (π_e) learns quickly, stays domain-specific

When entering a new domain:

1. Freeze μ: Strategic reasoning is already optimal
2. Reset π_e: Learn new motor skills from scratch
3. Adapt P, V: Fine-tune predictions for new dynamics

7 Experimental Predictions

7.1 Testable Hypotheses

H1: Transfer learning
An agent trained with emotional decomposition should demonstrate measurable transfer to new domains compared to training from scratch. The magnitude of this advantage is an empirical question.

H2: Ablation studies

- Removing emotion layer $\rightarrow$ performance should degrade significantly
- Using continuous emotions ($|\mathcal{E}| \rightarrow \infty$) $\rightarrow$ MCTS convergence should become problematic
- Varying $|\mathcal{E}| \rightarrow$ there should exist an optimal range balancing expressivity and computational efficiency

H3: Neuroscience parallels
Brain regions encoding emotional states (amygdala, insula) should exhibit:

- Discrete attractor dynamics (not continuous)
- Task-invariant activation patterns (transfer)
- Coordination with prefrontal cortex (MCTS-like planning)

7.2 Benchmark Tasks

Phase 1: Within-domain transfer

- Train on Minecraft survival → Transfer to Minecraft creative mode
- Train on easy FPS maps → Transfer to hard FPS maps

Phase 2: Cross-domain transfer

- Train on Minecraft → Transfer to Terraria (similar genre)
- Train on RTS games → Transfer to Tower Defense (strategic similarity)

Phase 3: Radical transfer

- Train on video games → Transfer to robotic manipulation
- Train on combat scenarios → Transfer to resource gathering

8 Comparison to Related Work

8.1 Hierarchical Reinforcement Learning

Options Framework (Sutton et al., 1999):

- Learns temporal abstractions bottom-up
- Our approach: Top-down emotional priors enable immediate transfer

Feudal Networks (Dayan & Hinton, 1993):

- Manager sets goals, worker executes
- Our approach: Emotion sets strategic stance, reflex executes

Key difference: We explicitly design for transfer by separating universal strategy (emotions) from domain-specific execution (reflexes).

8.2 Meta-Learning

MAML (Finn et al., 2017):

- Learns initialization for fast adaptation
- Requires gradient-based updates in new domains

Our approach:

- Zero-shot transfer of strategic reasoning
- Only reflex layer requires training

Advantage: No meta-training dataset needed—emotions are universal primitives.

8.3 World Models

Ha & Schmidhuber (2018):

- Learn latent dynamics for model-based RL
- Plan in latent space using continuous actions

Our approach:

- Plan in emotional space using discrete tokens
- Reduces branching factor from ∞ to $|\mathcal{E}|$

8.4 Intrinsic Motivation

Curiosity-driven exploration (Pathak et al., 2017):

- Single intrinsic reward (prediction error)
- No explicit strategic reasoning

Our approach:

- Multiple emotional tokens encode different exploration strategies
- MCTS balances them adaptively

9 Implementation Considerations

9.1 Computational Efficiency

MCTS complexity: $O(|\mathcal{E}|^h)$ per decision

With $|\mathcal{E}| = 8$ and $h = 5$: $8^5 = 32{,}768$ nodes (tractable)

With continuous actions and $h = 5$: ∞^5 nodes (intractable)

Parallelization: MCTS simulations are embarrassingly parallel—linear speedup with multiple cores.

Inference time:

- Encoder forward pass: $\sim$1ms
- MCTS search (100 iterations): $\sim$10–50ms
- Reflex actor forward pass: $\sim$1ms
- Total: $<$100ms per decision (real-time capable)

9.2 Training Pipeline

Stage 1: Single-domain training

1. Train encoder ϕ and reflex actor π_e jointly
2. Train world model P and value V from collected trajectories
3. Run MCTS at inference time (no training—statistics emerge naturally)

Stage 2: Transfer to new domain

1. Keep frozen: MCTS statistics (as starting point), emotional token set $\mathcal{E}$
2. Fine-tune: Encoder ϕ (different observations)
3. Retrain from scratch: Reflex actor π_e (different actions)
4. Adapt: World model P, value V (different dynamics)

The degree of transfer advantage depends on the similarity of strategic structures between source and target domains.

9.3 Challenges and Open Questions

Q1: Encoder alignment

How to ensure latent spaces z across domains remain aligned so μ transfers?

Possible solution: Contrastive learning with domain-invariant features, or explicit alignment losses.

Q2: Emotion granularity

Is $|\mathcal{E}|$ universally sufficient, or do some domains require finer distinctions?

Possible solution: Hierarchical emotions (coarse $\rightarrow$ fine) or adaptive token sets.

Q3: Continuous vs. discrete actions

Our reflex actor outputs continuous actions, but how to handle truly discrete domains (chess, Go)?

Possible solution: Emotion-conditioned policy networks that output discrete distributions.

10 Philosophical Implications

10.1 The Necessity of "Artificial Emotions"

Our framework suggests a provocative conclusion: General intelligence may require emotion-like structures not as optional features, but as computational necessities.

The pursuit of "purely rational" AI—maximizing expected utility through exhaustive search—is mathematically infeasible in continuous action spaces with sparse rewards. Emotions provide the discrete tokenization that makes strategic reasoning tractable.

This inverts the traditional view:

- Classical AI: "Emotions are irrational; remove them for optimal performance"
- Our view: "Emotions are the compression scheme that enables rationality"

10.2 Implications for AI Safety

If artificial general intelligence inherently requires emotion-like token systems, this has profound implications:

Alignment: We must align not just reward functions, but emotional response patterns.

Interpretability: Emotional tokens provide natural "explanation units"—an AI can report "I chose caution because..." rather than inscrutably optimizing a black-box value function.

Control: Emotional overrides (e.g., "always be cautious around humans") may be more robust than reward shaping.

11 Conclusion

We have presented a practical architecture for general intelligence based on discrete emotional tokens as transferable policy primitives. By decomposing decision-making into:

1. Universal strategy: Emotion-MCTS that transfers across domains
2. Domain-specific execution: Reflex actors that implement emotional directives

we enable efficient transfer learning while maintaining the computational tractability required for long-horizon planning.

Our key contributions:

Theoretical: Formalizing why emotion-action separation enables transfer learning in continuous control.

Architectural: Specifying the encoder-MCTS-actor pipeline with clear module boundaries.

Practical: Demonstrating that discrete emotional tokens suffice for general intelligence when combined with rich latent state representations.

The notion that "emotions make you irrational" may be precisely backwards. Emotions are the discrete tokens that make rationality computationally possible.

Future work should focus on:

- Large-scale empirical validation across diverse domains
- Neuroscientific testing of discrete vs. continuous emotional representations
- Extension to multi-agent settings where emotions coordinate social behavior

If our hypothesis holds, the path to artificial general intelligence may require not eliminating emotions, but understanding them as the universal language of strategic thought.

References

[1] Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2), 181–211.

[2] Silver, D., et al. (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676), 354–359.

[3] Ha, D., & Schmidhuber, J. (2018). World models. arXiv preprint arXiv:1803.10122.

[4] Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. ICML.

[5] Pathak, D., et al. (2017). Curiosity-driven exploration by self-supervised prediction. ICML.

[6] Damasio, A. R. (1994). *Descartes' Error: Emotion, Reason, and the Human Brain*. Putnam.

[7] Ekman, P. (1992). An argument for basic emotions. *Cognition & Emotion*, 6(3-4), 169–200.

[8] Baker, B., et al. (2022). Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. NeurIPS.