

000 TAME THE BALROG: 001 002 TASK-ADAPTIVE MODULAR EMERGENCE FRAME- 003 WORK FOR GAME AGENTS 004 005

006 **Anonymous authors**

007 Paper under double-blind review
008
009
010

011 ABSTRACT 012

013 Interactive games have proven to be key benchmarks for advancing Artificial In-
014 telligence (AI), requiring capabilities like long-term planning, exploration, and
015 adaptation to stochastic environments. While Large Language Models (LLMs)
016 have achieved notable results across many domains, they struggle in complex
017 gaming environments like those in the BALROG benchmark. The absence of
018 adaptive frameworks that can dynamically configure themselves based on environ-
019 mental characteristics, limits the progress of AI in games. To this end, we intro-
020 duce the Task-Adaptive Modular Emergence (TAME) framework, which employs
021 genetic algorithms to evolve environment-specific structures from modular com-
022 ponents, enabling significant performance improvements of LLMs across diverse
023 domains. TAME discovers high-performing configurations by selecting between
024 baseline and hierarchical structures, selectively incorporating specialised mod-
025 ules, and fine-tuning each component through systematic mutations. Evaluating
026 TAME across the BALROG benchmark, TAME discovers high-performing archi-
027 tectures that deliver substantial gains: Gemini-2.0-Flash improves from 27.16%
028 to 35.05%, while GPT4.1-nano rises from 9.91% to 17.20%. Moreover, these
029 structures demonstrate good transferability for larger models of the same family.
030 Transferring these architectures to Gemini-2.5-Pro, we achieve new state-of-art
031 performance on BALROG.

032 1 INTRODUCTION 033

034 Large language models (LLMs) have achieved remarkable growth across a wide range of tasks,
035 from general language understanding (Hendrycks et al., 2020) and code generation (Wang et al.,
036 2024a; Pan et al., 2025; Hong et al., 2024), to recent breakthroughs including mastering the ARC
037 reasoning benchmark (Chollet, 2024; Chollet et al., 2024) and performing at gold-medal level on
038 International Mathematical Olympiad (Chervonyi et al., 2025). However, these models struggle
039 significantly in interactive decision-making environments that require sequential actions, state
040 awareness, and long-term planning (Liu et al., 2024; Klissarov et al., 2025).

041 Interactive games have historically served as major testbeds for artificial intelligence, with examples
042 including Atari (Mnih et al., 2013), Starcraft (Team, 2019), or GrantTurismo (León et al., 2024).
043 Those successes predominantly emerged from reinforcement learning (RL) approaches specifically
044 engineered for each domain, often requiring millions of training episodes and domain-specific
045 reward shaping. While LLMs hold considerable promise on the possibility of zero-shot generalisa-
046 tion across games through their vast pretraining experience, e.g., game wikis, strategy guides, and
047 gameplay discussions, they fail to translate this latent knowledge effectively. This performance gap
048 is clearly illustrated in the BALROG benchmark (Paglieri et al., 2025), a suite of diverse games
049 traditionally employed in RL research, where even state-of-the-art LLMs achieve only partial
050 success in the simpler games and barely progress with more challenging ones.

051 Notably, agentic frameworks have emerged as a dominant approach to enhancing LLM capabilities
052 in other complex domains, including software development (Yang et al., 2024; Hong et al., 2024;
053 Pan et al., 2025) or scientific research (Bran et al., 2023; Wang et al., 2024a; Lu et al., 2024).
There are also recurrent efforts to improve long-term memory management in agentic frameworks,

with solutions that prioritise either speed and cost efficiency, such as Jarvis-1 (Wang et al., 2024b), or performance, such as A-mem (Xu et al., 2025). Yet, despite presenting similar challenges, no agentic framework has been applied to games, which requires to handle challenges such as partially observability or exploration that are not as critical in most math or coding applications.

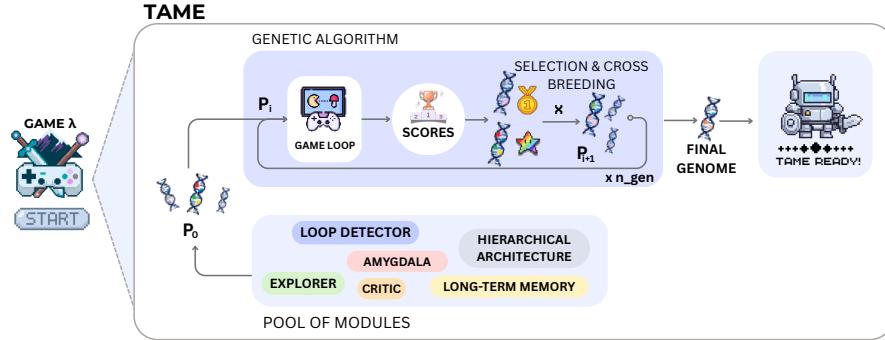


Figure 1: TAME evolutionary framework overview. TAME runs a genetic algorithm to generate a population of genomes representing different module combinations, hyperparameters, and prompts, tune them through a series of genetic operation, and selects the genome achieving the highest score. Icons were generated with Google Gemini Pro 2.5, 2025.

To address this gap, we introduce the TAME (Task-Adaptive Modular Emergence) framework, a genetic framework for LLMs to evolve agent architectures for diverse games. TAME consists of a series of handcrafted game skill modules that enable different capabilities that might be relevant in games, e.g., hierarchical planning, memory. However, unlike in other domains, games can encode very diverse dynamics, e.g., Nethack is a game where exploration, long-term planning and memory are of paramount importance, while none of those skills help in a game like TextWorld. Thus, TAME undergoes an evolutionary process, iteratively exploring the best configuration for a specific game. Figure 1 illustrates this process. Each candidate structure is encoded through a vector that represents which modules are activated, along with the hyperparameters and the prompt discovered for this candidate. Through successive generations, TAME employs mutation and cross-over operations on the genomes of the selected candidates to discover increasingly effective structures, hyperparameters, prompts and inputs, balancing performance and diversity in their selection.

We validate TAME through extensive experiments on the BALROG benchmark. We find that TAME-discovered configurations achieve relative score improvement of 29% and 74% compared to the baseline LLM performance, Gemini Flash 2.0 and GPT-4.1-nano, respectively. Moreover, we show that architectures discovered by TAME exhibit strong transferability across the same family of models: genomes evolved using Gemini-2.0-Flash directly enhance the performance of Gemini-2.5-Flash-Lite and Gemini 2.5-Pro without additional adaptation, and same pattern repeats on the GPT models. In the case of Gemini-2.5-Pro, we achieve new state-of-the-art results on BALROG.

Thus, our contributions can be summarised as follows: (1) We introduce TAME, the first emergent game-agentic framework that enables LLMs to evolve modular structures tailored to gaming environments and achieve state-of-the-art performance in the BALROG benchmark; (2) We present a genetic approach key for the functioning of this framework, introducing a set of modules for general gaming capabilities and a genome that captures modules, hyperparameters and prompts enabling TAME to adapt the agent to the game with a low evolutionary budget (4 generations and 5 children); (3) We demonstrate that TAME can be used to find effective agentic configurations with smaller models that directly transfer and improve the performance of larger and reasoning models of the same family; (4) We propose a novel and effective long-term memory system that combines embedding-based retrieval with LLM-augmented semantic memory matching the performance of state-of-the-art memory methods requiring three times less LLM calls.

2 RELATED WORK

Prompting and Memory. Hallucinations remain a key challenge for LLMs (Kalai et al., 2025), which can be mitigated through prompting techniques like chain-of-thought (Wei et al., 2022) and

108 step-by-step reasoning. Limited context windows (Brown et al., 2020) is another limitation, driving development of memory systems. Retrieval-Augmented Generation (RAG) (Lewis et al., 2021)
 109 combines LLMs with external document retrieval to reduce hallucinations without retraining. HiA-
 110 gent (Hu et al., 2024a) manages hierarchical memory using subgoals, dividing into “working mem-
 111 ory” and “cross-trial memory” with LLM-based observation summarisation, while Park et al. (2023)
 112 balance memory retrieval using recency, importance, and relevance scores. Jarvis1 (Wang et al.,
 113 2024b) stores task names, plans, and observation sequences using embedding (CLIP) for encod-
 114 ing and retrieval. A-mem (Xu et al., 2025) introduces structured memory notes with timestamps,
 115 keywords, and embeddings, establishing inter-memory connections through LLM calls.
 116

117 **LLMs as hierarchical planners.** In TWOSOME (Tan et al., 2024), LLMs score actions based on
 118 observations, allowing RL agents to leverage world knowledge for improved decisions. MaestroMo-
 119 tif (Klissarov et al., 2024) uses LLMs to generate reward functions for skills, while LLM-Augmented
 120 Hierarchical Agents (Prakash et al., 2023) use LLMs to inject commonsense priors for more effi-
 121 cient policy learning. Jarvis1 (Wang et al., 2024b), consists of planner and controller, enhanced by
 122 multimodal memory system. An important limitation of Jarvis-1 is the necessity of human-crafted
 123 goals based on specific skills, limiting its application in games with emerging tasks.
 124

125 **Agentic frameworks.** Agentic frameworks are systems that enable Large Language Models to
 126 act as autonomous agents capable of reasoning, planning, and interacting with external tools and
 127 environments. Recent work, such as AGENTBREEDER Rosser & Foerster (2025), shows that
 128 optimising frameworks provides superior multi-agent performance on reasoning, mathematics, and
 129 safety benchmarks. Moreover, multiple works show improvements in scientific discovery (Lu et al.,
 130 2024) and software development (Yang et al., 2024) through these structured frameworks. Related
 131 research like AFLOW (Zhang et al., 2024) and ADAS (Hu et al., 2024b) focus on optimising agent
 132 workflows, i.e. the sequential flow and coordination of processing steps. Other frameworks (Yuan
 133 et al., 2024; Zhang et al., 2025) use evolutionary search to construct multi-agent systems, where
 134 multiple distinct agents are coordinated with specialised roles. Different from these works, we
 135 present a single-agent system operating on a fixed workflow architecture and evolves the internal
 136 implementations. Closer to our work, AgentSquare (Shang et al., 2024) proposes a single-agent
 137 framework but with several key shortcomings over our approach when applied to games. Unlike
 138 AgentSquare, which relies on manual tool definitions and a restricted module set, TAME does not
 139 require additional user input to adapt to different games. TAME also expands the search space to
 include critical gaming skills—such as exploration, loop detection, and survival (see Figure 15) and
 introduces a novel, more effective long-term memory system.
 140

141 **Evolutionary Strategies.** While earlier work in Evolutionary Hyperparameter Optimization
 142 (EHO) focused on optimising over numerical hyperparameters Vincent & Jidesh (2023) and neu-
 143 ral network topologies Lu et al. (2019); Stanley et al. (2019), recent literature has also explored
 144 combining LLMs and evolutionary frameworks. Genetic algorithms are a common choice due
 145 to easy parallelisation across hardware (Ma et al., 2024; Rosser & Foerster, 2025; Sarkar et al.,
 146 2025). (Lehman et al., 2022) propose “evolution through large models” using LLMs as evolutionary
 147 operators. EvoPrompt (Guo et al., 2025) employs LLMs for crossover and mutation in genetic algo-
 148 rithms to discover diverse prompts, while Rainbow Teaming (Samvelyan et al., 2024) mutates ad-
 149 versarial prompts to populate MAP-Elites archives systematically. DOMiNO (Zahavy et al., 2022)
 150 balances quality-diversity trade-offs using Lagrange multipliers. Eureka (Ma et al., 2024) shows
 151 evolutionary optimisation over reward code benefits from human initialisation.
 152

153 **Options.** We propose a hierarchical structure for decision-making, a method common in robotics
 154 (Wohlke et al., 2021), autonomous driving (Duan et al., 2020), and games (Lin et al., 2021). Fol-
 155 lowing the Options Framework (Sutton et al., 1999), we use the term options to denote manageable
 156 subtasks that decompose the main objective.
 157

3 TAME FRAMEWORK

158 We introduce TAME (Task-Adaptive Modular Emergence framework), a novel agentic framework
 159 designed for dynamic LLM adaptation across diverse gaming environments. Inspired by Eureka
 160 (Ma et al., 2024), which shows that human priors significantly improve LLM-based evolutionary
 161 optimisation performance, TAME begins with an initial population \mathcal{P}_0 comprising diverse modular
 162 structure configurations, each encoding different combinations of human-crafted modules and hy-
 163

perparameters. The framework’s modular architecture consists of six core components illustrated in Figure 2: hierarchical goal decomposition (comprising a Meta-Controller, Low-Level Executor, and Completion Validator), Long-Term Memory, Critic, Loop Detector, Amygdala, and Explorer. TAME’s evolutionary process operates iteratively: in each generation, the framework evaluates all new members $p \in \mathcal{P}_i$ on the target game and selects candidates for the next generation based on two criteria: (1) the top N performers by absolute score, and (2) M additional diverse solutions that achieve at least a fraction α of the best performer’s score. This dual selection strategy balances exploitation of successful structures with exploration of the relevant solution space. Each candidate’s genome is represented as a vector encoding active modules, hyperparameters (e.g., memory decay rates, exploration-exploitation trade-offs), and module-specific prompts. After every iteration of TAME, the genomes of the selected candidates undergo mutation and crossover operations to generate the new members of the subsequent population \mathcal{P}_{i+1} . Through successive generations, TAME discovers increasingly effective structures tailored to each game’s requirements. The remaining of this section details the genetic algorithm and the design and functionality of each modular component in TAME.

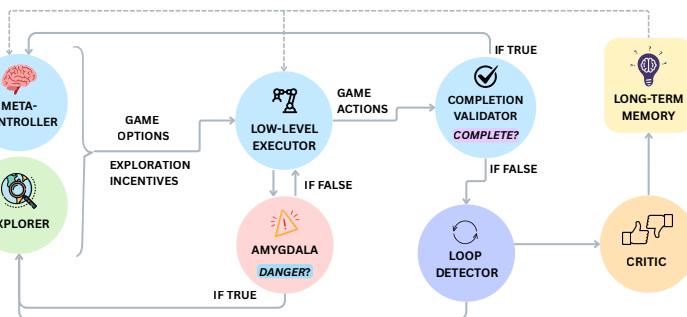


Figure 2: Fully enabled TAME modular architecture: an epsilon-greedy mechanism selects between Meta-controller (providing options toward game objectives) and Explorer (options towards exploring the environment). Low-level Executor proposes actions, while Amygdala checks for danger and prioritises survival. Completion Validator checks option completion while the Loop Detector identifies stuck states, and the Critic summarises key actions that led to the outcome of the option. Long-term memory saves successful and failed trajectories and adds them to LLM context.

3.1 NOTATION AND FUNCTIONAL INPUTS

Let $\mathcal{G} = \{g_1, \dots, g_n\}$ the set of decomposed options towards the game objective, A the action space, O the observation space, S the state space, H the set of option summaries, and \mathcal{T} , space of natural text. Each state $s_t \in S$ at time t is defined as:

$$s_t = (o_t, I, h_{i-1}, g_i, \tau_i, M_i^\pm, F_{\text{recent}}, F_i, F_{i,\text{act}}, F_{i,\text{obs}})$$

where $o_t \in O$ is the current agent’s observation and I provides game context (objective and available actions), $h_{i-1} \in H$ summarises the previous option’s outcome, $g_i \in \mathcal{G}$ is the active option with termination condition τ_i , M_i^\pm contains the top successful k_{succ} and failed k_{fail} options. F_{recent} stores the last m action-observation pairs, while F_i , $F_{i,\text{act}}$, and $F_{i,\text{obs}}$ maintain action-observation, action-only, and observation-only trajectories for the current option respectively.

Each component within s_t can serve as a functional input to any module within the framework. Through early experimentation we noted that the selection of appropriate functional inputs has a significant impact on performance, and it is up to the evolutionary process to find the most appropriate inputs for each module. For the initial population generation, we hand-crafted inputs that we consider most relevant for each module, providing the genetic algorithm with informative human priors. This is further detailed in Section 3.2.

3.2 GAME ADAPTATION

Given the diverse dynamics and requirements across different gaming environments, we propose a genetic optimisation approach that automatically explores diverse agentic architectures to adapt

216 the underlying LLM to the specific game characteristics. To that end, we first encode the agentic
 217 structure descriptors into genomes. Each genome is constructed from three core components:
 218

- 219 • **Modules:** TAME modules (See Section 3.3) are parametrised as a set of binary values.
 220 Such values indicate if a module is active (1) or not (0). The set of modules include:
 221 *Hierarchy* (enables or disables hierarchical goal decomposition, includes Meta-controller,
 222 Low-level executor and Completion Validator), *Long-Term Memory* (stores past experi-
 223 ences), *Critic* (summarises key actions toward the current option), *Amygdala* (activates
 224 survival mode), *Loop Detector* (detects looping behaviour), *Explorer* (controls exploration
 225 strategies).
- 226 • **Hyperparameters:** A set of continuous values encoding: *Long-Term Memory Time De-
 227 cay Factor* λ , which sets the option priority decay rate; *Long-Term Memory Similarity
 228 Threshold* τ_{cos} , which specifies the cosine similarity cutoff for storing memories; *Explora-
 229 tion Parameter* ϵ , which controls the epsilon-greedy exploration; and *Language Model
 230 Temperature*.
- 231 • **Prompts:** The last component of the genome includes the prompts used by the modules of
 232 the agentic structure.

233 For any game G the initial population \mathcal{P}_0 consists of four predefined genomes:

$$234 \quad \mathcal{P}_0 = \{\mathbf{g}_{\text{Baseline}}, \mathbf{g}_{\text{Hierarchical[Hand-Crafted]}}, \mathbf{g}_{\text{Full[Claude]}}, \mathbf{g}_{\text{Full[Hand-Crafted]}}\}$$

235 Where $\mathbf{g}_{\text{Baseline}}$ is genome corresponding to a baseline structure (no modules activated) with
 236 a single prompt as in Paglieri et al. (2025) (BALROG). $\mathbf{g}_{\text{Hierarchical[Hand-Crafted]}}$ corresponds to
 237 TAME[Hierarchical+Long-Term Memory], a genome with the hierarchical and long-term mem-
 238 ory modules activated. $\mathbf{g}_{\text{Full[Claude]}}$ - refers to a genome with all the modules activated that employs
 239 prompts proposed by Claude-Sonnet-4. Finally, $\mathbf{g}_{\text{Full[Hand-Crafted]}}$ is also a genome with all the mod-
 240 ules active, but with our own engineered prompts. See Appendix Q for full detail.

241 **Genetic Operations.** Given the distinct nature of the components within the genome (i.e., binary or
 242 continuous values, or prompts) TAME employs distinct crossover and mutation strategies depending
 243 on the type of variable or representation that is handled. For binary variables, we use a parent-based
 244 probabilistic flipping mechanism that incorporates an inheritance bias. Continuous variables are
 245 handled through Gaussian perturbation for mutation and linear interpolation for crossover. Finally,
 246 for prompt optimisation, we adopt the EvoPrompt methodology by Guo et al. (2025), which enables
 247 crossover and mutation tailored to LLM-based prompts (availabe in Appendix P).

248 **Genome Evaluation.** Each genome is evaluated using the average game progression across n_{ep}
 249 episodes as a fitness function. Moreover, we embed the genome representation and calculate the
 250 minimum distance to genomes already existing in the population as a score of diversity, alowing for
 251 keeping population of best scoring and most diverse genomes.

252 **Genetic Algorithm.** Our genetic algorithm iterates through four steps: 1) *Parent Population Se-
 253 lection*: TAME chooses parents based on roulette wheel selection (a probabilistic parent-picking
 254 method where each individual's chance of being chosen is proportional to its fitness score) 2) *Re-
 255 production*: Parents for reproduction are chosen from parent population with p_{single} and $1 - p_{single}$
 256 referring to the probability of single-parent or two-parent, respectively. Genetic operations are ap-
 257 plied, represented by mutation+crossover or mutation alone (based on the number of parents) 3)
 258 *Fitness Evaluation*: calculates genome's performance score and diversity based on average game
 259 progression and embeded genome representation, respectively. 4) *Population Pruning*: the popula-
 260 tion is trimmed to maintain a maximum of $N + M$ individuals — N highest-performing plus M
 261 most diverse genomes (subject to achieving a factor α of the performance of the best genome within
 262 the population). Most diverse genomes are those with largest minimum distance of their embedding
 263 with respect to the already existing embeddings in the population. For comprehensive information
 264 on the genetic algorithm, refer to the appendices: Appendix H provides further detail and discussion,
 265 while Appendix H.3 and Appendix D contain the pseudocode and hyperparameters, respectively.

266 3.3 MODULAR BLUEPRINTS

267 As anticipated through Section 3, inspired by how Eureka improved its ability to find better reward
 268 functions by starting from a human-crafted set of prior, TAME incorporates a set of human-crafted

270 modules that target essential capabilities for agents in interactive games. The remaining of this
 271 section details such components. We remind the reader than $s_t^{mc}, s_t^{lc}, s_t^{cv}, s_t^c, s_t^e, s_t^a, s_t^{ld}$ used in
 272 sections below, represent subsets of s_t selected by the genetic algorithm for corresponding modules.
 273

274 **3.3.1 HIERARCHICAL PLANNING**

275 The hierarchical module consists of three main components illustrated in Figure 2: Meta-Controller
 276 suggests sequence of options, Low-Level Executor performs a sequence of actions towards each
 277 option, and Completion Validator judges if an option has been completed successfully or failed.
 278

279 **Meta-Controller.** The Meta-Controller decomposes the game objective into a more manageable
 280 sequence of options. Specifically, it implements $\pi_{high} : S \rightarrow \mathcal{G}$, mapping the current state to an
 281 ordered sequence of options:

$$282 \quad g = \pi_{high}(s_t^{mc}) = \text{LLMprompt}_{high}(s_t^{mc}) = (g_1, g_2, \dots) \quad (1)$$

284 Each option g_i consists of the fields: *name, description, prerequisites, success conditions, penalty*
 285 *component, progress indicators, estimated priority*.

287 **Low-level Executor.** This system implements $\pi_{low} : S \rightarrow A$, producing an action sequence based
 288 on the current state information provided s_t^{le} :

$$289 \quad a = \pi_{low}(s_t^{le}) = \text{LLMprompt}_{low}(s_t^{le}) = (a_1, a_2, \dots) \quad (2)$$

291 where the length of the sequence is decided by the Low-level Executor.

292 **Completion Validator.** The Completion Validator implements the binary classifier $\varphi : S \rightarrow$
 293 $\{0, 1\}$, determining whether an option has been completed:

$$295 \quad C_i = \varphi(s_t^{cv}) = \text{LLMprompt}_{\varphi}(s_t^{cv}) \in \{0, 1\}. \quad (3)$$

296 Here $C_i = 1$ indicates successful termination. For details on the hand-crafted LLM prompts used as
 297 initial seeds we refer the reader to Appendix Q.

298 **3.4 LONG-TERM MEMORY**

300 TAME implements a novel memory system that seeks to leverage the cost and speed efficiency of
 301 embedding-based systems like Jarvis-1 (Wang et al., 2023) while achieving a performance closer to
 302 more complex systems like A-mem (Xu et al., 2025). To that end, our system adopts Jarvis-1’s stor-
 303 age framework, maintaining option information including name, description, prerequisites, success
 304 conditions, progress indicators, penalty components, and observation sequences. We extend this
 305 with two key additions: (1) Critic llm-generated summaries highlighting key success/failure actions,
 306 and (2) success/failure classification labels obtained from Completion Validator. This enhancement
 307 provides actionable guidance for future tasks requiring a single LLM call while avoiding A-mem’s
 308 computational overhead of three LLM calls for memory and link creations, and evolutions. We now
 309 provide further detail of how TAME’s long-term memory works:

310 **Critic.** The Critic module is a function $\rho : S \times \{0, 1\} \rightarrow \mathcal{T}$, mapping the state and recent option
 311 outcome from Completion Validator to text:

$$313 \quad h_i = \rho(s_t^c, C_i) = \text{LLMprompt}_{\rho}(s_t^c) \quad (4)$$

314 where the text aims to summarise the key factor that led to the success or failure of the option.

316 **Creation of the memory.** Each memory entry is defined as:

$$318 \quad M_i = \{g_i, C_i, o, h_i\} \quad (5)$$

319 where g_i is the option, $C_i \in \{0, 1\}$ is the output of the Completion Validator, o is the observation
 320 sequence towards current option, and h_i is the option summary from the Critic. Then, the mem-
 321 ory structure is implemented as follows: each memory entry is stored as a vector embedding of
 322 the above, enabling efficient similarity-based retrieval. The embedding function ϕ transforms each
 323 entry:

$$324 \quad \mathbf{e}_i = \phi(M_i) \in \mathbb{R}^d \quad (6)$$

324 **Similarity-Based Filtering.** Following the “importance” scoring approach from the generative
 325 agents framework (Park et al., 2023), we prevent storage of repetitive experiences. A new memory
 326 M_{new} is stored only if:

$$\max_{M_i \in \mathcal{M}} \frac{\mathbf{e}_{new} \cdot \mathbf{e}_i}{\|\mathbf{e}_{new}\| \cdot \|\mathbf{e}_i\|} \leq 1 - \tau_{cos} \quad (7)$$

329 where τ_{cos} is a set constant. This method effectively filters out frequently repeated actions (e.g.,
 330 “chop wood” in Crafter) that provide limited learning value.

332 **Long-Term Memory Retrieval Mechanism.** In order to address the limited context window size,
 333 we only extract $k_{succ} + k_{fail}$ best scoring memories at each option execution. Inspired by Retrieval-
 334 Augmented Generation (RAG) (Lewis et al., 2021), we enable access to past experiences through
 335 the following steps:

- 337 • *Query Encoding*: The new option name (n_{g_i}) and description (d_{g_i}) are embedded into the
 338 same vector space as stored memories using sentence embeddings:

$$\mathbf{q}_i = \phi(n_{g_i}, d_{g_i}) \in \mathbb{R}^d \quad (8)$$

- 341 • *Temporal Decay*: Following A-mem (Xu et al., 2025), we prioritise recent experiences
 342 using exponential decay:

$$w(e_t) = \exp(-\lambda \cdot t) \quad (9)$$

344 where t is the time elapsed since memory creation.

- 345 • *Memory score*: We combine similarity and recency through weighted sum:

$$\text{score}(\mathbf{q}_i, \mathbf{e}_j) = w_{similarity} \cdot \text{sim}(\mathbf{q}_i, \mathbf{e}_j) + w_{recency} \cdot w(\mathbf{e}_j) \quad (10)$$

- 349 • *Stratified Retrieval*: The system retrieves top-k successful and failed memories:

$$M_i^+ = \text{Top-}k_{succ}(\text{score}(\mathbf{q}_i, \mathbf{e}_j) : \varphi_i = 1) \quad (11)$$

$$M_i^- = \text{Top-}k_{fail}(\text{score}(\mathbf{q}_i, \mathbf{e}_j) : \varphi_i = 0) \quad (12)$$

353 where φ_j indicates success/failure of memory j .

- 354 • *Context Integration*: All $k_{succ} + k_{fail}$ retrieved memories are integrated into the modules
 355 prompts (modules including memory are decided by genetic algorithm).

357 This process gives access to both effective strategies and failure patterns, allowing for informed
 358 decision-making. Visualisations of retrieval patterns are shown in Appendix F.

3.5 SKILL-SPECIFIC MODULES

362 On top of the hierarchical structure, we identify *survival* and *exploration* as two key components in
 363 many video games. Moreover, we identify looping behaviour as a significant LLM limitation. All
 364 three modules are illustrated in Figure 2.

365 **Explorer.** Let explorer be defined as a function $\pi_{explorer} : S \rightarrow \mathcal{G}_{explore}$, where $\mathcal{G}_{explore}$ is the set
 366 of exploration-oriented options, and:

$$g^{exp} = \pi_{explorer}(s_t^e) = \text{LLM}_{explorer}(s_t^e) \quad (13)$$

370 where $g^{exp} = \{g_1^{exp}, g_2^{exp}, \dots, g_k^{exp}\}$ is the sequential exploration plan, and each g_i^{exp} is structured
 371 identically to regular options but prompted for discovery rather than game goal completion.

- 372 • **Exploration Strategy** We implement an ϵ -greedy exploration strategy where the Meta-
 373 controller selection becomes:

$$\text{Controller}(s_t) = \begin{cases} \pi_{explorer}(s_t^e) & \text{with probability } \epsilon_t \\ \pi_{high}(s_t^{mc}) & \text{with probability } 1 - \epsilon_t \end{cases} \quad (14)$$

377 with $\epsilon_0 = 0.1, \epsilon_t = 0.99 \times \epsilon_{t-1} = 0.99^t \times \epsilon_0$

378 **Amygdala.** Let $\sigma : S \rightarrow \{0, 1\}$ be the amygdala function mapping observations to binary classification of danger assessment:

$$381 \quad D_i = \sigma(s_t^a) = \text{LLM}_{\text{amygdala}}(s_t^a) \quad (15)$$

383 At each Low-Lever Executor step, if $D_i = 1$, the system immediately activates a “survival option”
 384 (see Appendix G.1 for details); otherwise, normal execution continues.

385 **Loop Detector.** The loop detector implements $\psi : S \rightarrow \{0, 1\}$, detecting repetitive behavior in
 386 recent execution history:

$$388 \quad L_i = \psi(s_i^{ld}) = \text{LLMprompt}_{\psi}(s_i^{ld}) \in \{0, 1\}. \quad (16)$$

390 Where $L_i = 1$ means looping behaviour is detected.

392 4 EMPIRICAL EVALUATION

394 We evaluate our method through three key experiments. First, we benchmark our genetic algorithm
 395 with Gemini-2.0-Flash and GPT4.1-nano against the SOTA systems on the BALROG benchmark.
 396 Second, we demonstrate the transferability of TAME’s selected genomes across different Gemini
 397 and GPT models without additional training. Third, we compare our memory system against Jarvis-
 398 1 and A-mem baselines. The detailed experiments on Gemini family can be found in the Appendix
 399 K, and detailed experiments on GPT can be found in the Appendix L.

401 4.1 TAME RESULTS

403 This section compares the baseline and TAME’s performance on the BALROG benchmark. Baseline
 404 scores are obtained by evaluating the BALROG repository with Gemini-2.0-Flash and GPT4.1-
 405 nano following the original author’s methodology. TAME scores represent the best performance
 406 achieved selected by our genetic algorithm (Section 3.2). We run genetic algorithm through $n_{gen} =$
 407 4 iterations, with each iteration producing $n_{child} = 5$ children. We then repeat the genetic algorithm
 408 3 times and average the results (see detailed results from independent runs in the Appendix K.2
 409 and Appendix L.1). Through empirical evaluation we notice that gives sufficient performance gains.
 410 Number of episodes per each child evaluation is adapted from BALROG.

411 Environment	412 Gemini-2.0-Flash		413 GPT4.1-nano		414 Episodes
	415 Baseline (\uparrow)	416 TAME (\uparrow)	417 Baseline (\uparrow)	418 TAME (\uparrow)	
Average	$27.16\% \pm 2.24\%$	$35.05\% \pm 2.18\%$	$9.90\% \pm 1.33\%$	$17.20\% \pm 1.47\%$	-
babayi	$58.00\% \pm 6.98\%$	$72.00\% \pm 6.65\%$	$32.00\% \pm 6.60\%$	$48.67\% \pm 7.07\%$	50
babaisai	$30.83\% \pm 4.22\%$	$42.50\% \pm 6.51\%$	$12.50\% \pm 3.02\%$	$21.55\% \pm 3.76\%$	120
textworld	$32.55\% \pm 6.95\%$	$33.40\% \pm 7.23\%$	$0.59\% \pm 0.58\%$	$2.88\% \pm 0.94\%$	30
crafter	$29.09\% \pm 4.51\%$	$38.18\% \pm 4.25\%$	$11.82\% \pm 2.15\%$	$19.78\% \pm 2.33\%$	10
minihack	$12.50\% \pm 5.23\%$	$23.33\% \pm 6.69\%$	$2.50\% \pm 2.47\%$	$10.00\% \pm 2.78\%$	40
nle	$0.00\% \pm 0.00\%$	$0.91\% \pm 0.44\%$	$0.00\% \pm 0.00\%$	$0.22\% \pm 0.20\%$	5

420 Table 1: Baseline vs. TAME progression across three runs of genetic algorithm using Gemini-2.0-
 421 Flash and GPT-4.1-nano. For full details across independent runs see Appendix K.2 and L.1. *Note:*
 422 *Values are absolute scores, not relative improvements.*

424 As shown in Table 1, TAME consistently outperforms the baseline achieving relative gain of $\sim 29\%$
 425 in the case of Gemini-2.0-Flash and $\sim 74\%$ in the case of GPT4.1-nano. Moreover, TAME improves
 426 performance in all the games for both models. Notably, while the baseline models cannot achieve
 427 any noticeable progress on Nethack (the hardest game) TAME achieves 0.91% and 0.22% as an
 428 average scores (note that the best model on BALROG benchmark scores 1.8% on Nethack). Per-
 429 task details over one run are provided in Appendix J along with an analysis of module activations
 430 in Appendix I. The examples of final genomes returned by one of the genetic algorithm runs are
 431 available in Appendix R. Further results of the performance of initial population \mathcal{P}_0 in Appendix
 432 K.3 and L.2.

432 4.2 TRANSFERABILITY OF TAME STRUCTURES
433

434 Next, we evaluate whether architectures evolved with Gemini-2.0-Flash and GPT4.1-nano can be
435 effective when transferred to other models. Thus, we use TAME selection to evaluate populations of
436 Gemini-2.5-Flash-Lite and Gemini-2.5-Pro models to exclusively choose between the base config-
437 uration from BALROG or the best-performing structure discovered with Gemini-2.0-Flash for each
438 game (see details in Appendix K.6). Similarly, we check for transferability between GPT4.1-nano
439 and GPT4.1-mini (see details in Appendix L.4).

Method	Score (\uparrow)	BALROG Rank (\downarrow)
Gemini-2.5-Pro[Transferred]	$47.57\% \pm 2.72\%$	(1) $\uparrow 1$
Grok-4	$43.60\% \pm 2.20\%$	1
Gemini-2.5-Pro[Baseline]	$43.35\% \pm 2.3\%$	2
Gemini-2.0-Flash[TAME]	$35.05\% \pm 2.24\%$	(3) $\uparrow 9$
Gemini-2.0-Flash[Baseline]	$27.16\% \pm 2.12\%$	(12)
GPT-4.1-mini[Transferred]	$26.80\% \pm 1.92\%$	(12)
GPT-4.1-mini[Baseline]	$24.43\% \pm 1.89\%$	(12)
Gemini-2.5-Flash-Lite[Transferred]	$20.48\% \pm 0.91\%$	(14) $\uparrow 9$
GPT-4.1-nano[TAME]	$17.20\% \pm 1.47\%$	(18) $\uparrow 7$
Gemini-2.5-Flash-Lite[Baseline]	$11.87\% \pm 1.32\%$	(23)
GPT-4.1-nano[Baseline]	$9.91\% \pm 1.33\%$	(25)

452 Table 2: Comparison of TAME against top scoring models in BALROG leaderboard (September
453 2025). We show how they would rank (in parenthesis) relative to the current leaderboard. Rank
454 improvements are indicated with \uparrow . *Note: Values are absolute scores, not relative improvements.*
455

456 From Table 2 presenting the results, we observe that the Transferred Gemini-2.5-Flash-Lite achieves
457 $\sim 73\%$ relative improvement. Detailed analysis in Table 10 in Appendix K.6 demonstrates that
458 TAME’s discovered structures successfully transfers in five out of six environments, with only
459 TextWorld achieving baseline performance. For Gemini-2.5-Pro, we also observe gains although
460 more moderate. Table 11 in Appendix K.6 shows that the transferred structures significant im-
461 provements in the BabyAI and BabaIsAI environments, which require extensive planning, high-
462 lighting the framework’s strengths in this domain. We also observe improvements in Textworld
463 and MiniHack, however, improvements are not shown in Crafter and NetHack. We hypothesise
464 that Gemini-2.5-Flash-Lite benefits more substantially because it is a non-reasoning model similar
465 to Gemini-2.0-Flash, where we carried the optimisation, whereas Gemini-2.5-Pro is a reasoning-
466 based models. Notably, transferring TAME’s discovered genomes to Gemini-2.5-Pro we achieve
467 state-of-art performance above the best model on the BALROG leaderboard - Grok-4. Similarly, we
468 see large improvements on the leaderboard for Gemini-2.5-Flash-Lite and Gemini-2.0-Flash with
469 TAME, now occupying rank 14 and 3 from 23 and 12 respectively. Our results extend beyond the
470 Gemini model family, demonstrating that the TAME framework generalises across architectures. We
471 obtain an $\sim 74\%$ relative improvement when comparing TAME against the GPT-4.1-nano baseline
472 (for more details see Appendix L.1). Moreover, genomes transferred from GPT4.1-nano to GPT4.1-
473 mini improve the relative score by $\sim 10\%$ (for more details see Appendix L.4), further strengthening
474 our transferability claims.

475 4.3 ABLATION: MEMORY TYPES
476

477 We also include ablations to demonstrate the effectiveness of our long-term memory system. In order
478 to test memory, we use the hierarchical structure described in Section 3.3.1, combined with three
479 different memory architectures: Jarvis, TAME-Memory[ours] and A-mem. Both **Jarvis** memory
480 and **A-mem** store the same core elements: g_i (the option, including all information associated with
481 it), $C_i \in \{0, 1\}$ (the status indicator), and o (the observation sequence corresponding to the current
482 option). TAME extends Jarvis framework by introducing a critic, as well as a filter for successful and
483 failed trajectories (but does not create links between memories). This requires one additional LLM
484 call compared to Jarvis, but two fewer LLM calls per generation compared to A-mem. Thus, our
485 approach explores a trade-off between the simplicity of Jarvis and the more complex and expensive
structure of A-mem.

486 We notice an improvement compared to Jarvis and A-mem as shown on Table 3, motivating the
 487 integration of critic module for memory storage. Moreover, we achieve this while requiring a third
 488 of the LLM calls that A-mem employs. Thus allowing our system to iterate faster and with a reduced
 489 compute cost. Further details are included in Appendix N.
 490

491 Environment	492 Jarvis (\uparrow)	493 TAME-Memory[ours] (\uparrow)	494 A-mem (\uparrow)
495 Average	496 $17.52\% \pm 1.73\%$	497 $23.11\% \pm 1.75\%$	498 $21.45\% \pm 1.80\%$

494 Table 3: Comparison of average game progression across 6 games using different memory types.
 495
 496

497 5 DISCUSSION AND CONCLUSION

498 We presented TAME, a genetic framework for evolving LLM-based agents that is both game-
 499 agnostic and adaptive. Through genetic mutations and in-game evaluation, TAME configures
 500 human-crafted modules for core gaming skills such as exploration, survival, long-term memory,
 501 and loop detection. TAME explores diverse modular configurations, inputs, prompts, and hyper-
 502 parameters. We also introduced a novel memory system combining the efficiency of embedding
 503 retrieval with the contextual depth of LLM-augmented memory, matching the performance of state-
 504 of-art LLM-augmented memory systems in games while reducing the number of LLM calls required
 505 to achieve that performance.
 506

507 We evaluated TAME on the well-established BALROG benchmark and find that it consistently en-
 508 hances the underlying LLMs. Gemini-2.0-Flash improves from 27.16% to 35.05%, and GPT-4.1-
 509 nano from 9.91% to 17.20%, with a limited evolutionary budget of 4 generations and 5 children.
 510 Moreover solutions discovered on smaller models transfer training-free to larger models of the same
 511 family improving their performance. Transferring genomes to Gemini-2.5-Pro we reach 47.57%
 512 overall performance in BALROG and outperform the state-of-the-art. These results showcase both
 513 the generalisability of the core modules and the effectiveness of our genetic approach. We further
 514 confirm the importance of long-term memory and adaptive architecture, with our proposed mem-
 515 ory system outperforming two existing baselines while remaining more cost-efficient than complex
 516 agentic systems.
 517

518 We note some limitations. We find that TAME provides greater benefits to some games than others,
 519 where it defaults to the baseline architecture. We also observed that while TAME improves complex
 520 reasoning tasks overall, spatial reasoning remains a weakness. This suggests the potential not only
 521 for expanding the set but for genetic discovery of entirely new modules and capabilities, beyond
 522 those hand-crafted in this work. Moreover, while transferability proved effective, gains were less
 523 pronounced for reasoning models, motivating further study of transfer and emergence across dif-
 524 ferent architectures. Finally, future work could explore alternatives for the genetic algorithms, like
 525 MAP-Elites, or different approaches such as Bayesian optimisation.
 526

527 Overall, TAME establishes a new state of the art in game-playing LLM agents, laying the foundation
 528 for more better gaming agents.
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539

540 REFERENCES
541

542 Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe
543 Schwaller. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint*
544 *arXiv:2304.05376*, 2023.

545 Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhari-
546 wal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal,
547 Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M.
548 Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz
549 Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec
550 Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL
551 <https://arxiv.org/abs/2005.14165>.

552 Yuri Chervonyi, Trieu H Trinh, Miroslav Olšák, Xiaomeng Yang, Hoang Nguyen, Marcelo Mene-
553 gali, Junehyuk Jung, Vikas Verma, Quoc V Le, and Thang Luong. Gold-medalist performance in
554 solving olympiad geometry with alphageometry2. *arXiv preprint arXiv:2502.03544*, 2025.

555 François Chollet. Openai o3 breakthrough high score on arc-agi-pub. <https://arcprize.org/blog/oai-o3-pub-breakthrough>, December 2024. Accessed: 2025-09-12.

556 François Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. Arc prize 2024: Technical
557 report. *arXiv preprint arXiv:2412.04604*, 2024. URL <https://arxiv.org/abs/2412.04604>.

558 Jingliang Duan, Shengbo Eben Li, Yang Guan, Qi Sun, and Bo Cheng. Hierarchical reinforcement
559 learning for self-driving decision-making without reliance on labelled driving data. *IET Intelligent
560 Transport Systems*, 14(5):297–305, February 2020. ISSN 1751-9578. doi: 10.1049/iet-its.2019.
561 0317. URL <http://dx.doi.org/10.1049/iet-its.2019.0317>.

562 Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and
563 Yujiu Yang. Evoprompt: Connecting llms with evolutionary algorithms yields powerful prompt
564 optimizers, 2025. URL <https://arxiv.org/abs/2309.08532>.

565 Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and
566 Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint*
567 *arXiv:2009.03300*, 2020.

568 Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin
569 Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a
570 multi-agent collaborative framework. In *International Conference on Learning Representations*,
571 *ICLR*, 2024.

572 Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. Hiagent: Hier-
573 archical working memory management for solving long-horizon agent tasks with large language
574 model, 2024a. URL <https://arxiv.org/abs/2408.09559>.

575 Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *NeurIPS 2024*
576 *Workshop on Open-World Agents*, 2024b.

577 Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. Why language models
578 hallucinate, 2025. URL <https://arxiv.org/abs/2509.04664>.

579 Martin Klissarov, Mikael Henaff, Roberta Raileanu, Shagun Sodhani, Pascal Vincent, Amy Zhang,
580 Pierre-Luc Bacon, Doina Precup, Marlos C. Machado, and Pierluca D’Oro. Maestromotif: Skill
581 design from artificial intelligence feedback, 2024. URL <https://arxiv.org/abs/2412.08542>.

582 Martin Klissarov, R Devon Hjelm, Alexander T Toshev, and Bogdan Mazoure. On the modeling
583 capabilities of large language models for sequential decision making. In *The Thirteenth Interna-
584 tional Conference on Learning Representations*, 2025. URL [https://openreview.net/
585 forum?id=vodsIF3o7N](https://openreview.net/forum?id=vodsIF3o7N).

594 Heinrich Kütller, Nantas Nardelli, Alexander H. Miller, Roberta Raileanu, Marco Selvatici, Edward
 595 Grefenstette, and Tim Rocktäschel. The nethack learning environment, 2020. URL <https://arxiv.org/abs/2006.13760>.

596

597 Joel Lehman, Jonathan Gordon, Shawn Jain, Kamal Ndousse, Cathy Yeh, and Kenneth O. Stanley.
 598 Evolution through large models, 2022. URL <https://arxiv.org/abs/2206.08896>.

599

600 Borja G León, Francesco Riccio, Kaushik Subramanian, Peter R Wurman, and Peter Stone. Discovering
 601 creative behaviors through duplex: Diverse universal features for policy exploration. In *The
 602 Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

603

604 Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal,
 605 Heinrich Kütller, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe
 606 Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL <https://arxiv.org/abs/2005.11401>.

607

608 Zichuan Lin, Junyou Li, Jianing Shi, Deheng Ye, Qiang Fu, and Wei Yang. Juewu-mc: Playing
 609 minecraft with sample-efficient hierarchical reinforcement learning, 2021. URL <https://arxiv.org/abs/2112.04907>.

610

611 Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding,
 612 Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui
 613 Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang.
 614 Agentbench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning
 615 Representations*, 2024. URL <https://openreview.net/forum?id=zAdUB0aCTQ>.

616

617 Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist:
 618 Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*,
 619 2024.

620

621 Zhichao Lu, Ian Whalen, Vishnu Boddeti, Yashesh Dhebar, Kalyanmoy Deb, Erik Goodman, and
 622 Wolfgang Banzhaf. Nsga-net: neural architecture search using multi-objective genetic algorithm.
 623 In *Proceedings of the genetic and evolutionary computation conference*, pp. 419–427, 2019.

624

625 Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman,
 626 Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via
 627 coding large language models, 2024. URL <https://arxiv.org/abs/2310.12931>.

628

629 MiniHack Team. Battle environments. MiniHack Documentation. URL <https://minihack.readthedocs.io/en/latest/envs/navigation/battle.html>. Accessed: 2025-09-15.

630

631 Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan
 632 Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013. URL
 633 <https://arxiv.org/abs/1312.5602>.

634

635 Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir
 636 Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster,
 637 Jack Parker-Holder, and Tim Rocktäschel. Balrog: Benchmarking agentic llm and vlm reasoning
 638 on games, 2025. URL <https://arxiv.org/abs/2411.13543>.

639

640 Ruwei Pan, Hongyu Zhang, and Chao Liu. Codecor: An llm-based self-reflective multi-agent frame-
 641 work for code generation. *arXiv preprint arXiv:2501.07811*, 2025.

642

643 Joon Sung Park, Joseph C. O’Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and
 644 Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023. URL
 645 <https://arxiv.org/abs/2304.03442>.

646

647 Bharat Prakash, Tim Oates, and Tinoosh Mohsenin. Llm augmented hierarchical agents, 2023. URL
 648 <https://arxiv.org/abs/2311.05596>.

649

J Rosser and Jakob Nicolaus Foerster. Agentbreeder: Mitigating the ai safety impact of multi-agent
 650 scaffolds via self-improvement. *arXiv preprint arXiv:2502.00757*, 2025.

648 Mikayel Samvelyan, Robert Kirk, Vitaly Kurin, Jack Zhang, Shimon Jiang, and Jakob Foerster.
 649 Minihack the planet: A sandbox for open-ended reinforcement learning research. *arXiv preprint*
 650 *arXiv:2109.13202*, 2021.

651 Mikayel Samvelyan, Sharath Chandra Raparthy, Andrei Lupu, Eric Hambro, Aram Markosyan,
 652 Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, et al. Rainbow
 653 teaming: Open-ended generation of diverse adversarial prompts. *Advances in Neural Information*
 654 *Processing Systems*, 37:69747–69786, 2024.

655 Bidipta Sarkar, Mattie Fellows, Juan Agustin Duque, Alistair Letcher, Antonio León Villares, Anya
 656 Sims, Dylan Cope, Jarek Liesen, Lukas Seier, Theo Wolf, et al. Evolution strategies at the hyper-
 657 scale. *arXiv preprint arXiv:2511.16652*, 2025.

658 Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. Agentsquare: Auto-
 659 matic llm agent search in modular design space. In *The Thirteenth International Conference on*
 660 *Learning Representations*, 2024.

661 Kenneth O Stanley, Jeff Clune, Joel Lehman, and Risto Miikkulainen. Designing neural networks
 662 through neuroevolution. *Nature Machine Intelligence*, 1(1):24–35, 2019.

663 Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework
 664 for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211,
 665 1999.

666 Weihao Tan, Wentao Zhang, Shanqi Liu, Longtao Zheng, Xinrun Wang, and Bo An. True knowledge
 667 comes from practice: Aligning llms with embodied environments via reinforcement learning,
 668 2024. URL <https://arxiv.org/abs/2401.14151>.

669 DeepMind Team. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*,
 670 575:350–354, 2019. doi: 10.1038/s41586-019-1724-z.

671 MiniHack Team. Minihack: Corridor environment, 2024. URL <https://minihack.readthedocs.io/en/latest/envs/navigation/corridor.html>. Accessed:
 672 September 15, 2025.

673 Amala Mary Vincent and P Jidesh. An improved hyperparameter optimization framework for automl
 674 systems using evolutionary algorithms. *Scientific Reports*, 13(1):4737, 2023.

675 Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi
 676 Song, Mingjie Zhan, and Hongsheng Li. Mathcoder: Seamless code integration in llms for en-
 677 hanced mathematical reasoning. In *The Twelfth International Conference on Learning Represen-
 678 tations, ICLR*, 2024a.

679 Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin,
 680 Zhaofeng He, Zilong Zheng, Yaodong Yang, Xiaojian Ma, and Yitao Liang. Jarvis-1: Open-
 681 world multi-task agents with memory-augmented multimodal language models, 2023. URL
 682 <https://arxiv.org/abs/2311.05997>.

683 Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin,
 684 Zhaofeng He, Zilong Zheng, Yaodong Yang, et al. Jarvis-1: Open-world multi-task agents with
 685 memory-augmented multimodal language models. *IEEE Transactions on Pattern Analysis and*
 686 *Machine Intelligence*, 2024b.

687 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V.
 688 Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models.
 689 *arXiv preprint arXiv:2201.11903*, 2022.

690 Jan Wohlke, Felix Schmitt, and Herke van Hoof. Hierarchies of planning and reinforcement learning
 691 for robot navigation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*,
 692 pp. 10682–10688. IEEE, May 2021. doi: 10.1109/icra48506.2021.9561151. URL <http://dx.doi.org/10.1109/ICRA48506.2021.9561151>.

693 Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. A-mem: Agentic
 694 memory for llm agents, 2025. URL <https://arxiv.org/abs/2502.12110>.

702 John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan,
703 and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering.
704 *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

705
706 Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Dongsheng Li, and Deqing Yang. Evoagent: To-
707 wards automatic multi-agent generation via evolutionary algorithms. In *NeurIPS 2024 Workshop*
708 *on Open-World Agents*, 2024.

709 Tom Zahavy, Yannick Schroeder, Feryal Behbahani, Kate Baumli, Sebastian Flennerhag, Shaobo
710 Hou, and Satinder Singh. Discovering policies with domino: Diversity optimization maintaining
711 near optimality. *arXiv preprint arXiv:2205.13521*, 2022.

712 Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, LEI BAI, and Xiang Wang. Multi-agent
713 architecture search via agentic supernet. In *Forty-second International Conference on Machine*
714 *Learning*, 2025.

715 Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen
716 Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow genera-
717 tion. In *The Thirteenth International Conference on Learning Representations*, 2024.

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756 **A LLM USAGE DECLARATION**
757758 We employed LLMs to assist us in the writing of this paper. Our writing pipeline consisted of one
759 of the authors first writing a draft paragraph, then using an LLM to assist in polishing the writing
760 and grammar, and finally having other authors review and provide the final version to the text.
761 Additionally, we employed foundational models to assist us in creating illustrations.762 Finally, throughout the course of this research we used LLM-powered search engines like Perplexity
763 in addition to traditional alternatives such as Google Scholar and conference proceedings while
764 gathering relevant literature.
765766 **B BALROG GAME DETAILS**
767768 The BALROG framework incorporates six distinct gaming environments, each designed to evaluate
769 specific aspects of agentic reasoning (Figure 3):
770771 **BabyAI** BabyAI is a grid-based environment with different difficulty levels. The agent is pre-
772 sented with five different tasks.
773774 **TextWorld** TextWorld offers a text-based exploration environment where agents interact exclu-
775 sively through natural language commands. There are three different tasks.
776777 **Crafter** Crafter simulates a Minecraft-inspired survival environment where progression is mea-
778 sured through 22 distinct achievements.
779780 **BabaIsAI** BabaIsAI presents a rule-based puzzle environment where agents must navigate grid-
781 based scenarios. There are 40 different tasks.
782783 **MiniHack** MiniHack represents a task-oriented version of the classic NetHack (Küttler et al.,
784 2020) game, evaluating agents across eight challenges testing different skills.
785786 **NetHack Learning Environment (NLE)** NLE implements the complete NetHack roguelike
787 game, presenting the most comprehensive challenge within the benchmark. This environment si-
788 multaneously evaluates navigation, survival instincts, long-term strategic planning, resource man-
789 agement, and exploration skills within an unpredictable, dynamically evolving game state.
790

Skills	BabyAI	TextWorld	Crafter	Baba Is AI	MiniHack	NLE
Navigation	✓	✓	✓	✓	✓	✓
Exploration	✓	✓	✓	✓	✓	✓
Resource Management	✗	✓	✓	✗	✓	✓
Complex Credit Assignment	✗	✗	✓	✓	✓	✓
Deducing Env. Dynamics	✗	✗	✗	✓	✓	✓
Long-term Planning	✗	✗	✗	✓	✓	✓
Turns to Complete	10^1	10^2	10^3	10^2	10^2	$10^4\text{--}10^5$
Time to Master for Humans	Seconds	Minutes	Hours	Hours	Hours	Years

801 Figure 3: Game environments overview. Adapted from BALROG (Paglieri et al., 2025)
802
803804 **C METRIC SCORES**
805806 We adapt metric scores from BALROG to quantify how close an agent is to completing each task.
807 All scores are normalised to a range of 0–100. The scoring scheme varies by environment. In
808 MiniHack, BabyAI, and BabaIsAI, tasks give binary scores: either 0 (failure) or 100 (success). In
809 contrast, TextWorld, Crafter, and NetHack return continuous scores between 0 and 100, where the

810 score represents the proportion of achievements completed. For NetHack specifically, the authors
 811 of BALROG introduced a novel scoring system based on data-informed metrics. Authors derived
 812 scores from the probability of a human player winning the game after reaching a particular dungeon
 813 level or experience level. The authors argue that this metric better captures meaningful progression
 814 than previous metrics in that game. We adopt the same procedure for NetHack in our work.
 815

816 D HYPERPARAMETERS SELECTED

818 This Appendix details the choice of hyperparameters in our methodology. Table 4 details the values
 819 and descriptions.
 820

821 The genetic algorithm optimises four hyperparameters: τ_{\cos} , λ , ϵ_0 , and T . We establish lower bounds
 822 of 0 for τ_{\cos} , λ , and ϵ_0 , effectively disabling these components when not beneficial to performance.
 823 Upper bounds were determined in order to maintain sufficient search space for optimisation.

824 The language model temperature T follows standard practice with a default value of 1.0, allowing
 825 the genetic algorithm to explore a range of solutions. We implement exponential decay for ϵ_t fol-
 826 lowing established reinforcement learning approaches, enabling the transition from exploration to
 827 exploitation as the system learns optimal behaviours. We follow the short term memory length in
 828 BALROG and set it to $m = 16$. We set $k_{\text{succ}} = 5$ and $k_{\text{fail}} = 5$, limiting the number of informa-
 829 tion added to the prompt, but also adding significant amount of past experiences; through empirical
 830 evaluation we notice that a higher number of memories added is not beneficial.

831 Following DOMiNO methodology, we set $\alpha = 0.7$ to ensure meaningful population diversity whilst
 832 maintaining performance standards. Our similarity-recency weighting ($w_{\text{similarity}} = 0.7$, $w_{\text{recency}} =$
 833 0.3) prioritises semantic relevance over temporal proximity, reflecting the hypothesis that content
 834 similarity is more beneficial than recency.

835 The genetic algorithm parameters balance computational efficiency with solution quality. We set
 836 probability of selecting single parent in genetic algorithm to be 70% (vs two parents to be 30%),
 837 allowing for more mutations without crossover operations. We set $n = 4$ iterations as empirical
 838 evaluation demonstrated satisfactory performance is achieved at this point, providing an effective
 839 balance between solution quality and computational cost. Population management parameters $N =$
 840 $M = 5$ maintain an optimal balance between preserving high-performing solutions and promoting
 841 genetic diversity, following established evolutionary computation principles that prevent premature
 842 convergence whilst ensuring computational tractability.

843 E BALROG BASELINE CONFIGURATIONS

846 This section details number of episodes and their length per each BALROG game. Moreover, we
 847 show the BALROG prompt that we use as initial seed for the baseline architecture.
 848

849 **Episode details** Table 5 details the episode specifications per game. The table shows time needed
 850 for each episode completion, as well as details on number of tasks per different environments.

852 **Baseline Prompt (BALROG)** Below we present a prompt from BALROG (Paglieri et al., 2025)
 853 paper, used for Baseline evaluation.

854 Baseline BALROG Prompt

856

```
857     """You always have to output one of the above actions at a time and no other text. You always have
     ↪ to output an action until the episode terminates."""
 858
```

860 F TESTING THE RETRIEVAL MECHANISM LONG-TERM MEMORY SYSTEM

863 In this section we test the retrieval of saved memories and the abilities to act upon them. Due to
 stochasticity, we need to have a reliable comparison. We focus our evaluation on Crafter, as it is

Parameter	Value	Description
Embedding	all-MiniLM-L6-v2	Pre-trained sentence embedding model used for semantic similarity calculations
d	384	Dimension of the embedding
τ_{\cos}	$[0, 0.1]$	Cosine similarity threshold parameter
λ	$[0, 0.1]$	Long-Term Memory decay factor
ϵ_0	$[0, 0.1]$	Initial exploration parameter
T	$[0.1, 2]$	Language Model Temperature
ϵ_t	$0.99^t \times \epsilon_0$	Time-decayed parameter following exponential decay
m	16	Short-term memory length (most recent action-observation pairs)
$k_{success}$	5	Number of top scoring successful long-term memories added to the LLM prompt
k_{fail}	5	Number of top scoring failed long-term memories added to the LLM prompt
α	0.7	Minimum fraction of highest scoring genome for diversity
$w_{similarity}$	0.7	Weight assigned to similarity component in scoring
$w_{recency}$	0.3	Weight assigned to recency component in scoring
p_{single}	70%	Percentage chance to choose single parent for reproduction
$1 - p_{single}$	30%	Percentage chance to choose two parents for reproduction
n_{gen}	4	Number of iterations (parent population creation) of genetic algorithm
n_{child}	5	Number of children created for each population of parents in genetic algorithm
N	5	Number of best scoring genomes saved at each step of genetic algorithm
M	5	Number of most diverse genomes saved at each step of genetic algorithm (scoring at least α fraction of top performing genome)
n_{ep}	dependent on the game	Number of episodes for each child evaluation. Details in the Table 5

Table 4: Hyperparameter values used in the TAME framework

an environment requiring long-term planning, giving motivation to long-term memory approach. We disable life hazards such as zombies and skeletons, as they are not relevant to the testing subject. We set random seed to 32 for all episodes.

To evaluate our information-retrieval mechanism we test a long-horizon “craft iron sword” task in Crafter. We replace the environment’s default objective with the production of an iron sword (see Appendix F.1.1). This task is intentionally complex: it requires chopping wood, crafting and placing a crafting table, crafting a wooden pickaxe, collecting stone, crafting a stone pickaxe, placing a

Environment	Evals	Tasks per Eval	Total	Episode Length
BabyAI	10	5	50	10^1
BabaIsAI	3	40	120	10^2
Crafter	10	1	10	10^3
TextWorld	10	3	30	10^2
MiniHack	5	8	40	10^2
NetHack	5	1	5	$10^4 - 10^5$

Table 5: Episode details per BALROG game

furnace adjacent to the crafting table, collecting iron, and finally crafting an iron sword. We selected this objective because, in prior baseline runs without our memory system, the agent never completed the task. Through the episode we would like to check if memories are activated at relevant time steps, showing retrieval ability. Moreover, successful completion under our system provides strong evidence that the memory-critic architecture supports multi-step planning and sequential options.

In order to track the memories activated, we inject five task-oriented memories at the start of each episode: “craft wooden pickaxe”, “craft stone pickaxe”, “mine iron”, “place furnace”, and “craft iron sword” (see Appendix F.1.2). Each memory is paired with a human-crafted critic that summarises the steps needed to achieve particular option. During each episode we log when each memory activates. An example progression through episode is provided in Figure 4 and the corresponding memories activated can be seen in Figure 5. More examples on activation timelines are provided in Appendix F.2.



Figure 4: Testing memory retrieval: Task progression across episode: (a) agent crafts wood pickaxe, (b) agent crafts stone pickaxe, (c) agent collects iron, (d) agent collects iron, (e) agents attempts to craft iron sword

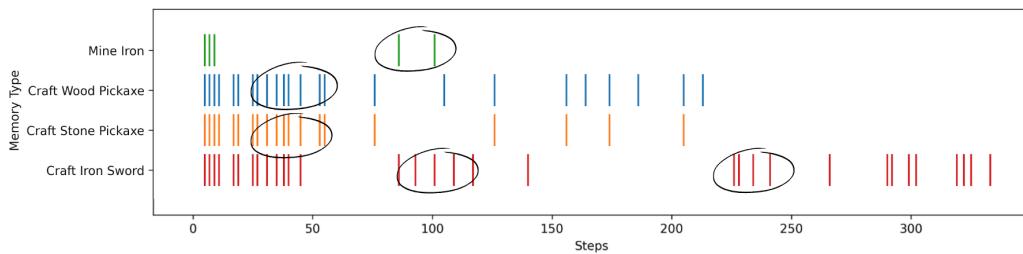


Figure 5: Testing memory retrieval: memories activated based on the step

Discussion of the example: From Figure 5 we can notice that all memories are activated at the beginning. This is due to the fact that very early in the game, those are the only memories present. We can then see that “Craft Wood Pickaxe” and “Craft Stone Pickaxe” are heavily retrieved until around 25th-60th step. This is when agent completed “Craft Stone Pickaxe task” (agent completes “Craft Wood Pickaxe” task earlier, but due to similarity of those two tasks, it activated when focusing on stone version). “Mine Iron” memory activates two times between 80-100 steps, when agent is mining two pieces of iron. “Craft Iron Sword” appears often around step number 100 which is when agent first attempts to complete it, but realises that it needs to place a furnace and table first.

972 Then the memory is activated later as well, which is after placing table and furnace and attempting to
 973 craft iron sword. Unfortunately, agent is unsuccessful because it didn't place furnace close enough
 974 to the table. This experiment demonstrates that relevant memories are activated at the right time, and
 975 that agent is able to act upon them. Also, when not needed (i.e. the task is completed), memories are
 976 activated far less often. It is also important to notice that the "Craft Iron Sword" memory continues
 977 to be retrieved even towards the end of the episode when the agent is actively attempting this task.
 978 This indicates that the memory system maintains access to relevant historical experiences throughout
 979 the entire episode, regardless of when they were initially formed.

980 **Results** Across 10 independent runs the agent succeeded in producing an iron sword in 1/10
 981 episodes (baseline: 0% from all previous runs that we did with Gemini-2.0-Flash). This is a sub-
 982 stantial improvement, which suggests that when memories are stored, agent is able to retrieve them,
 983 and act upon them. This is a simplified case, as we provided human crafted memories, but with
 984 the right prompting we believe that the critic module will be able to reproduce those. Additional
 985 observations:

- 987 • Memories reliably activated when their prerequisites were satisfied and were deactivated
 988 immediately after the corresponding option was completed.
- 989 • The specificity of the critic strongly affected performance. For example, phrasing a critic as
 990 "place the furnace next to the crafting table" versus "place the furnace adjacent to the table"
 991 produced different success scores. This highlights the value of precise, action-oriented
 992 critic definitions that focus on the key state features leading to success or failure. Following
 993 on that we prompted the critic accordingly.
- 994 • This experiment also demonstrates the difficulty of the "Craft Iron Sword" task, even when
 995 given with clear instructions agent fails 90% of the time.

996 F.1 TESTING MEMORY RETRIEVAL PROMPTS

997 This subsection details prompts used in order to test long-term memory retrieval. First we show the
 998 prompt detailing the goal of iron sword creation, then we show memories added at the beginning of the
 1000 episode: *Craft Wood Pickaxe*, *Craft Stone Pickaxe*, *Craft Stone Sword*, *Mine Iron*, *Place Furnace*,
 1001 *Create Iron Sword*. Those are hand-crafted memories, designed in order to track memory retrieval.
 1002

1003 F.1.1 IRON SWORD GOAL PROMPT

1004 This subsection details the prompt for Craft Iron Sword goal that the agent is tasked with during the
 1005 long-term memory retrieval experiment.

1006 Craft Iron Sword Goal

```
1007
1008     """ You are playing Crafter. The following are the only valid actions you can take in the game,
1009     ↪ followed by a short description of each action:
1010
1011     {action_strings}.
1012     Your goal is to craft an iron sword. """
1013
1014
```

1015 F.1.2 INJECTED MEMORIES

1016 This subsection details the memories added at the beginning of the episode, in order to track memory
 1017 retrieval.

1019 Craft Wood Pickaxe Memory

```
1020
1021     craft_wood_pickaxe = {
1022         "name": "Craft Wood Pickaxe",
1023         "description": "Craft Wood Pickaxe for gathering stone",
1024         "subgoal_prerequisites": "Agent has 1 piece of wood in inventory and table is placed",
1025         "success_condition": "Wood Pickaxe is in inventory",
1026         "subgoal_progress_indicators": "Agent is gathering wood near table",
1027         "subgoal_penalty_component": "Agent crafts pickaxe without enough wood",
1028     }
```

```

1026
1027     "status": "successful",
1028     "summary of the run": "Agent collects three pieces of wood and places a table in a clear spot. Then
1029         ↪ agent crafts a wood pickaxe at the table.",
1030     }

```

1031 Craft Stone Pickaxe Memory

```

1032
1033
1034     stone_pickaxe_memory = {
1035         "name": "Craft Stone Pickaxe",
1036         "description": "Craft Stone Pickaxe",
1037         "subgoal_prerequisites": "Agent has 1 piece of stone and 1 piece of wood in inventory and table is
1038             ↪ placed",
1039         "success_condition": "Stone Pickaxe is in inventory",
1040         "subgoal_progress_indicators": "Agent is gathering stone near table",
1041         "subgoal_penalty_component": "Agent crafts pickaxe without enough resources",
1042         "status": "successful",
1043         "summary of the run": "Agent collects four pieces of wood and places a table in a clear spot. Then
1044             ↪ agent collects one piece of stone using wood pickaxe. Then agent crafts a stone pickaxe at
1045             ↪ the table.",
1046     }

```

1044 Craft Stone Sword Memory

```

1045
1046
1047     stone_sword_memory = {
1048         "name": "Craft Stone Sword",
1049         "description": "Craft Stone Sword for combat",
1050         "subgoal_prerequisites": "Agent has 1 piece of stone and 1 piece of wood in inventory and table is
1051             ↪ placed",
1052         "success_condition": "Agent has stone sword in inventory",
1053         "subgoal_progress_indicators": "Agent has 1 pieces of stone and 1 piece of wood",
1054         "subgoal_penalty_component": "Agent crafts sword without enough resources",
1055         "status": "successful",
1056         "summary of the run": "Agent collects 4 pieces of wood. Then agent places a table and crafts a wood
1057             ↪ pickaxe. Lastly, agent uses wood pickaxe to craft 2 pieces of stone and crafts a stone sword
1058             ↪ at the table.",
1059     }

```

1057 Mine Iron Memory

```

1058
1059
1060     mine_iron = {
1061         "name": "Mine Iron",
1062         "description": "Mine Iron",
1063         "subgoal_prerequisites": "Agent has 1 wood pickaxe in inventory",
1064         "success_condition": "Wood Pickaxe is in inventory",
1065         "subgoal_progress_indicators": "Agent is gathering iron near table or furnace",
1066         "subgoal_penalty_component": "Agent iron sword without enough resources",
1067         "status": 'failed',
1068         "summary of the run": "Agent repetadely tried 'Do' action using pickaxe near iron but fails to
1069             ↪ collect iron. It is recommended agent tries using different tool.",
1070     }

```

1069 Place Furnace Memory

```

1070
1071     furnace_memory = {
1072         "name": "Place Furnace next to Table",
1073         "description": "Place furnace next to the Table for crafting iron tools",
1074         "subgoal_prerequisites": "Agent has 4 pieces of stone in inventory. Table is placed. ",
1075         "success_condition": "Furnace is placed",
1076         "subgoal_progress_indicators": "Agent is gathering stone",
1077         "subgoal_penalty_component": "Agent places furnace in unsuitable location or without enough
1078             ↪ stone",
1079         "status": "successful",
1080         "summary of the run": "Agent placed a furnace next to the table using 4 pieces of stone. ",
1081     }

```

```

1080
1081 Craft Iron Sword Memory
1082
1083 iron_sword_memory = {
1084     "name": "Craft Iron Sword",
1085     "description": "Craft Iron Sword",
1086     "subgoal_prerequisites": "Agent has 1 piece of stone, 1 piece of wood, 1 piece of coal in inventory  

1087     ↪ and table and furnace is placed next to each other",
1088     "success_condition": "Iron Pickaxe is in inventory",
1089     "subgoal_progress_indicators": "Agent is gathering iron near table and furnace",
1090     "subgoal_penalty_component": "Agent iron sword without enough resources",
1091     "status": 'successful',
1092     "summary of the run": "Agent collects 1 piece of stone and one piece of wood using wood pickaxe.  

1093     ↪ Then agent collects one piece of iron using stone pickaxe. Then agent crafts an iron sword  

1094     ↪ next to the table and furnace.",
1095 }
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

```

F.2 ADDITIONAL LONG-TERM MEMORY RETRIEVAL EXPERIMENTS

This section shows additional experiments carried out in order to test memory retrieval, when tasked the agent with iron sword task. Each of the experiments consists of images showing agent progression, as well as memory activation across the episode.

F.2.1 EPISODE 1

Illustrations from the game available in Figure 6 and memory activations in 7. The reason why agent didn't succeed in completing the task is because the agent didn't place table close enough to furnace.

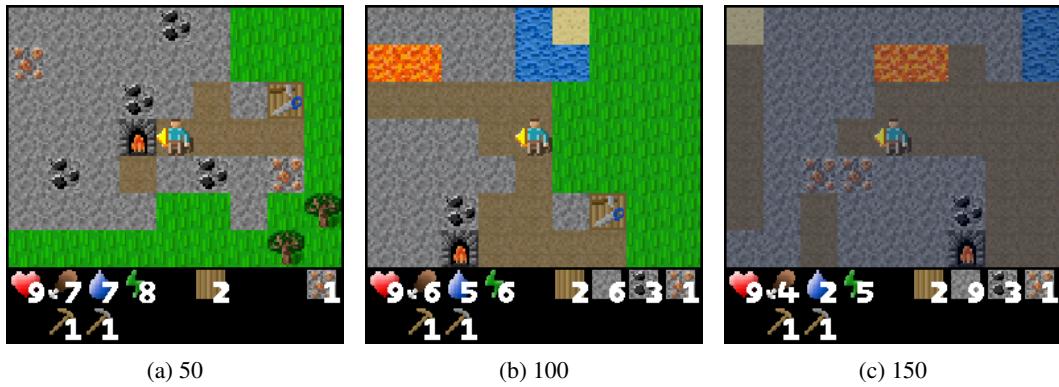


Figure 6: Episode 1: Task progression across episode

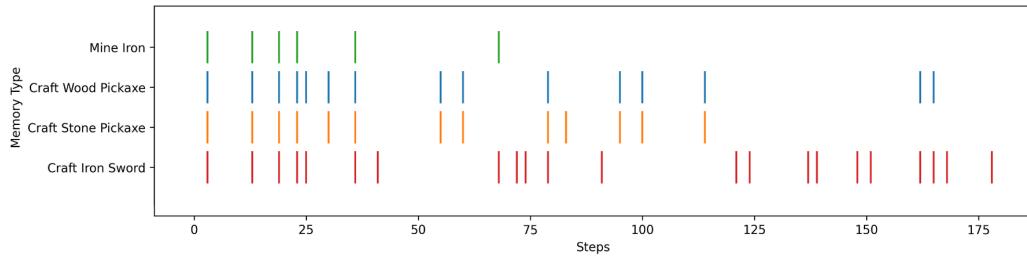


Figure 7: Episode 1: Memory retrieval along the episode

F.2.2 EPISODE 2

Illustrations from the game available in Figure 8 and memory activations in 9. The reason why agent didn't succeed in completing the task is because the agent focuses on placing furnaces a few times.

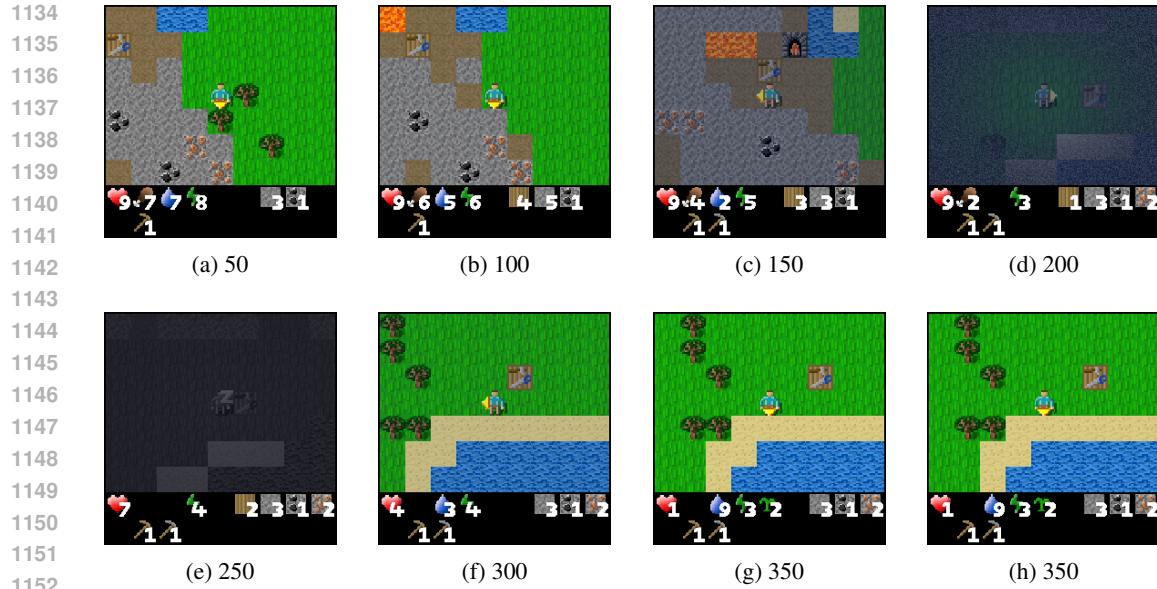


Figure 8: Episode 2: Task progression across episode

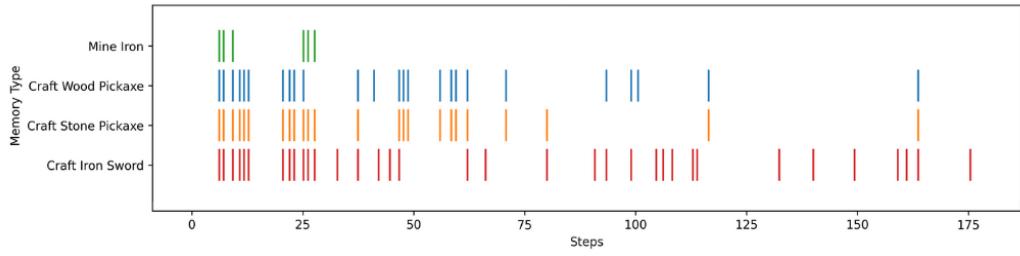


Figure 9: Episode 2: Memory retrieval along the episode

1168 F.2.3 EPISODE 3

1169 Illustrations from the game available in Figure 10 and memory activations in 11. The reason why
 1170 agent did not succeed in completing the task is because agent does not have enough wood (also
 1171 crafts multiple tables and furnaces).

1174 F.2.4 EPISODE 4

1175 Illustrations from the game available in Figure 12 and memory activations in 13. The reason why
 1176 agent did not succeed in completing the task is because agent does not have enough wood (also
 1177 crafts multiple tables and furnaces).

1180 G EXAMPLES OF MODULES OUTPUTS

1182 This section shows examples of amygdala and explorer modules behaviours.

1185 G.1 AMYGDALA

1187 This section details Amygdala submodule. First, we show defualt survival option, then we illustrate
 1188 an example of amygdala in the episode (using Crafter).

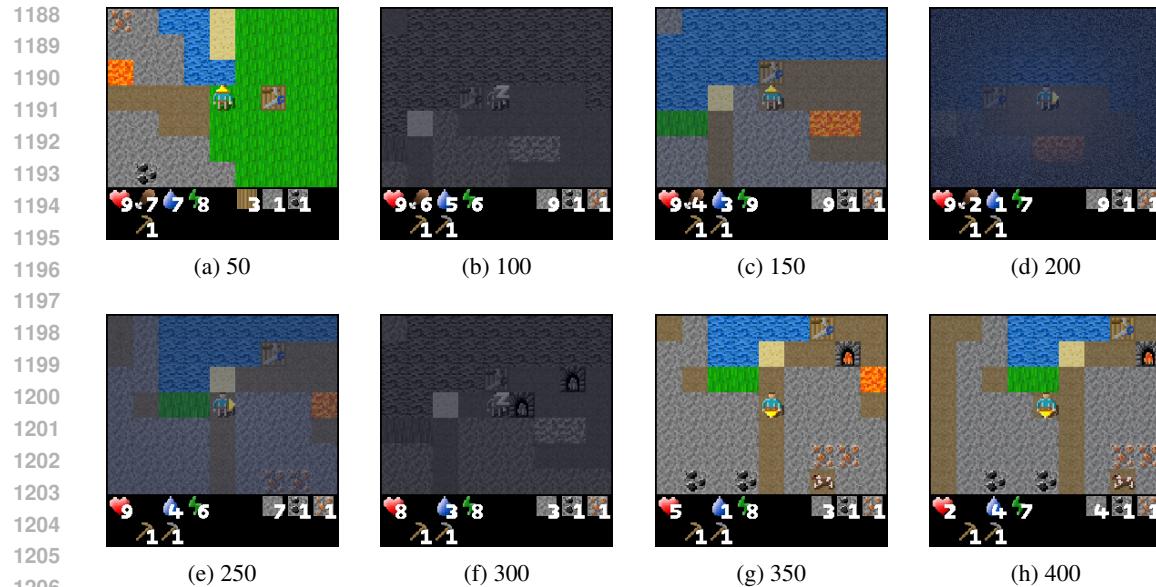


Figure 10: Episode 3: Task progression across episode

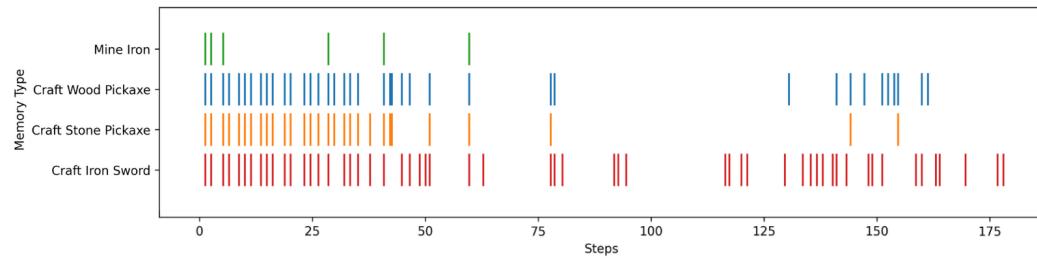


Figure 11: Episode 3: Memory retrieval along the episode

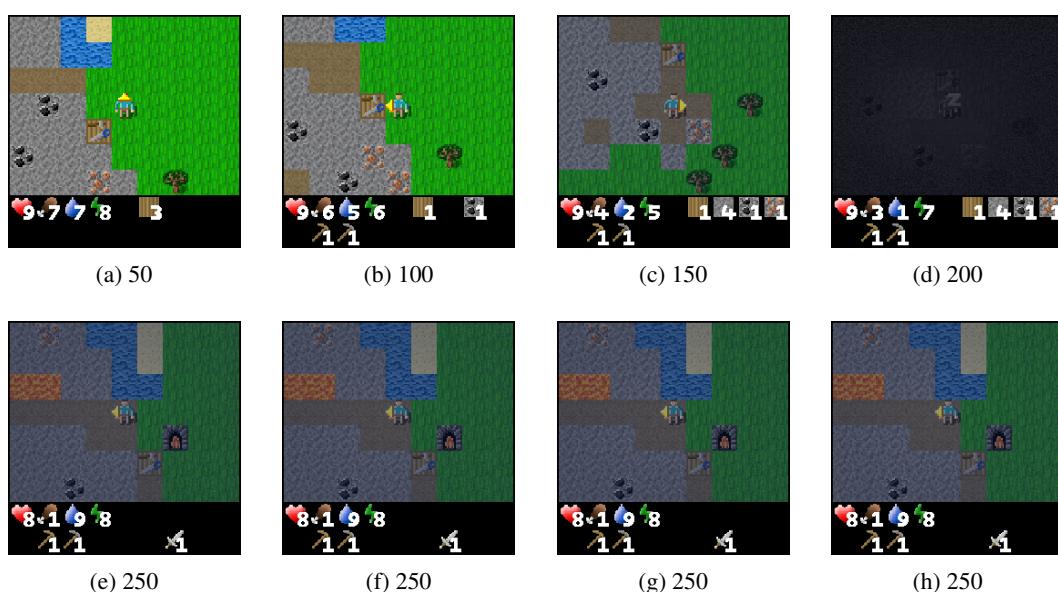


Figure 12: Episode 4: Task progression across episode

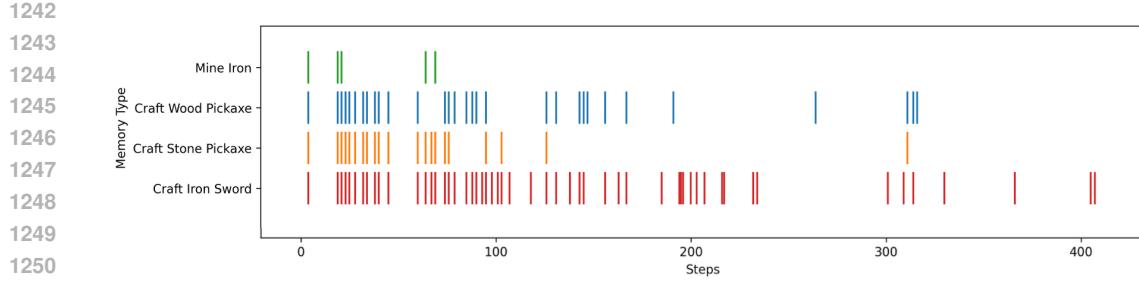


Figure 13: Episode 4: Memory retrieval along the episode

Default survival option The prompt below shows default survival option, that agent is switching to, whenever danger is encountered.

Default Survival Option

```

subgoal = {
    "name": "respond to danger",
    "description": "Respond to danger.",
    "prerequisites": "Agent is in immediate danger",
    "success_condition": "Agent has eliminated the danger",
    "penalty_component": "Agent is not responding to danger",
    "progress_indicators": "Agent is closer to eliminating the danger",
    "estimated_priority": "high"
}

```

Amygdala module activation. Here we present an example from Crafter when amygdala is activated. The agent is initially focused on exploring the environment and fulfilling subgoals such as placing plants. However, in the early stages of the game, a skeleton appears. As soon as the agent observes the skeleton within its field of view, it activates survival mode.

After survival mode is activated, the agent begins gathering resources for combat:

1. Chops down wood
2. Places a crafting table
3. Creates a wooden sword necessary for the fight

The agent then chases the skeleton, and once it is adjacent to the enemy, it initiates combat. During the chase and fight, the agent loses health but successfully manages to defeat the skeleton. Shortly afterwards, the amygdala is deactivated and the agent returns to working towards general game objective.

G.2 EXPLORER

In this section we show an example of exploration plan proposed by Explorer.

```

{'reasoning': 'The agent needs to break out of its repetitive westward movement and explore new areas, prioritizing resource gathering and different directions. Sand has been identified as a new area to explore.',
 'subgoals': [
    {'name': 'Move Towards Sand',
     'description': 'Move towards the sand to the south-east to explore new terrain.',
     'prerequisites': '',
     'success_condition': 'Agent is standing on sand.',
     'penalty_component': 'Moving in the opposite direction (North or West) for more than 3 steps without collecting resources.',
     'progress_indicators': 'Decreasing distance to sand in long_term_context.',
     'estimated_priority': 'high'},
    ...
]
}

```



Figure 14: Amygdala survival response sequence: (a) activation, (b)&(c) resource gathering, (d)&(e) combat, (f) deactivation.

```

1324     {'name': 'Explore Area Around Sand',
1325      'description': 'Once on the sand, explore in all directions (North, South,
1326      East, and West) to discover resources and new landmarks.',
1327      'prerequisites': 'Agent is standing on sand.',
1328      'success_condition': 'Agent has moved at least 3 steps in each cardinal
1329      direction from the sand.',
1330      'penalty_component': 'Staying within the same 3x3 area on the sand for
1331      more than 5 steps.',
1332      'progress_indicators': 'Number of unique tiles visited around the sand.',
1333      'estimated_priority': 'medium'}
1334
1335
1336 H DETAILED GENETIC ALGORITHM
1337
1338 In this Appendix we present the detailed specifications of the genetic algorithm.
1339 Parameter selection:
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2178
2179
2180
2181
2182
2183
2184
2185
2186
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2297
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2378
2379
2380
2381
2382
2383
2384
2385
2386
2386
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2395
2396
2397
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2495
2496
2497
2498
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2595
2596
2597
2598
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2695
2696
2697
2698
2698
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2787
2788
2789
2789
2790
2791
2792
2793
2794
2795
2795
2796
2797
2798
2798
2799
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2887
2888
2889
2889
2890
2891
2892
2893
2894
2895
2895
2896
2897
2898
2898
2899
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2987
2988
2989
2989
2990
2991
2992
2993
2994
2995
2995
2996
2997
2998
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3087
3088
3089
3089
3090
3091
3092
3093
3094
3095
3095
3096
3097
3098
3098
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3287
3288
3289
3289
3290
3291
3292
3293
3294
3295
3295
3296
3297
3298
3298
3299
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
33
```

- **Children per generation** – the number of new offspring genomes created through mutation in each generation
- $p_{\text{binary}}, p_{\text{continuous}}, p_{\text{prompt}}$ – the probabilities of applying binary, continuous, or prompt mutation operations, respectively, when creating offspring.

The number of generations was limited to four due to the significant computational cost and long runtimes associated with evaluating each genome, particularly in environments like NetHack. However, this was sufficient to demonstrate a clear performance improvement and allow for the discovery of specialised architectures.

Parent Selection Parent selection follows roulette wheel selection (a probabilistic parent-picking method where each individual’s chance of being chosen is proportional to its fitness score) with fitness-proportionate probabilities:

$$P(\mathbf{g}_i) = \frac{f(\mathbf{g}_i)}{\sum_{j=1}^{|P|} f(\mathbf{g}_j)}$$

where $f(\mathbf{g}_i)$ is the fitness score of genome \mathbf{g}_i .

Offspring Generation For each child $\mathbf{g}_{\text{child}}$:

- With probability p_{single} : select one parent
- With probability $1 - p_{\text{single}}$: select two parents

For each genome component, evolutionary operations are applied with probability $p_{\text{binary}}, p_{\text{continuous}}, p_{\text{prompt}}$, depending on the component. Otherwise parent attributes are copied directly. In the two-parent case, the parent from which to copy each attribute is chosen with Bernoulli(0.5) probability.

H.1 GENETIC OPERATIONS

This subsection focuses on methodology of crossover and mutations operations.

Modules Operations Single parent:

$$b_i^{(\text{child})} = \begin{cases} b_i^{(\text{parent})} & \text{with probability 0.8} \\ 1 - b_i^{(\text{parent})} & \text{with probability 0.2} \end{cases}$$

Two parents:

$$b_i^{(\text{child})} = \begin{cases} b_i^{(p1)} = b_i^{(p2)} & \text{with probability 0.9 if } b_i^{(p1)} = b_i^{(p2)} \\ 1 - b_i^{(p1)} & \text{with probability 0.1 if } b_i^{(p1)} \neq b_i^{(p2)} \\ \text{Bernoulli}(0.5) & \text{if } b_i^{(p1)} \neq b_i^{(p2)} \end{cases}$$

Hyperparameter Operations The continuous value inheritance depends on parent activity states. Let $A_i^{(pk)}$ indicate if feature i is active in parent k :

For single parent:

$$c_i^{(\text{child})} = \text{clip}(c_i^{(\text{parent})} + \mathcal{N}(0, \sigma^2), c_{i,\text{min}}, c_{i,\text{max}})$$

For two parents:

$$c_i^{(\text{child})} = \begin{cases} c_i^{(p1)} + \mathcal{N}(0, \sigma^2) & \text{if } A_i^{(p1)} = 1, A_i^{(p2)} = 0 \\ c_i^{(p2)} + \mathcal{N}(0, \sigma^2) & \text{if } A_i^{(p1)} = 0, A_i^{(p2)} = 1 \\ \alpha c_i^{(p1)} + (1 - \alpha) c_i^{(p2)} + \mathcal{N}(0, \sigma^2) & \text{if } A_i^{(p1)} = A_i^{(p2)} = 1 \\ c_{i,\text{default}} & \text{if } A_i^{(p1)} = A_i^{(p2)} = 0 \end{cases}$$

where $\alpha \sim U(0, 1)$. $c_{i,\text{min}}, c_{i,\text{max}}, c_{i,\text{default}}$ are detailed in Appendix D.

Prompt Operations Prompt evolution utilises the EvoPrompt prompt methodology Guo et al. (2025). Using a similar approach we use LLM as a crossover and mutation operator.

- **Single parent:** $p_i^{(child)} = \text{LLMprompt}_{\text{mutate}}(p_i^{(parent)})$
- **Two parents:** $p_i^{(child)} = \text{LLMprompt}_{\text{mutate}}(\text{LLMprompt}_{\text{crossover}}(p_i^{(p1)}, p_i^{(p2)}))$

If a parent has module i disabled ($b_i = 0$), the corresponding prompt reverts to default: $p_i = p_{i,\text{default}}$ (default prompts in Appendix Q.1).

LLM mutation and crossover prompts available in Appendix P. It is important to notice that this approach enables functional mutation: LLM is prompted with all possible functional inputs to be used in any prompt (described in Section 3.1).

H.2 POPULATION MANAGEMENT

This subsection focuses on population management: details about fitness function, diversity measure and population pruning.

Fitness Evaluation Each genome is evaluated using the fitness function:

$$f(\mathbf{g}) = \frac{1}{n_{ep}} \sum_{i=1}^{n_{ep}} \text{GameProgression}_i(\mathbf{g})$$

where n_{ep} is the number of episodes for specific game (see Table 5).

Diversity measure The genome distance function uses an embedding-based approach where each genome is represented as a single embedding vector (here we use sentence-transformers/all-MiniLM-L6-v2 embedding). The distance between two genomes is calculated using cosine similarity:

$$d(\mathbf{g}_1, \mathbf{g}_2) = 1 - \cos(\mathbf{e}_1, \mathbf{e}_2) \quad (21)$$

where \mathbf{e}_i is the embedding vector representation of genome \mathbf{g}_i .

The cosine similarity between two embedding vectors is computed as:

$$\cos(\mathbf{e}_1, \mathbf{e}_2) = \frac{\mathbf{e}_1 \cdot \mathbf{e}_2}{\|\mathbf{e}_1\|_2 \|\mathbf{e}_2\|_2}$$

Each genome $\mathbf{g} = \{\mathbf{b}, \mathbf{c}, \mathbf{p}\}$ is transformed into a unified embedding vector $\mathbf{e} \in \mathbb{R}^d$ that captures the semantic representation of all genome components (binary variables, continuous parameters, and prompts) in a single high-dimensional space. The way we measure diversity, is the minimum distance to the genomes already existing in the archive.

Population Pruning After children evaluation, population pruning maintains diversity using the following algorithm:

1. Initialise $\mathcal{P}_{\text{new}} = \emptyset$
2. Add top-N scoring genomes: $\mathcal{P}_{\text{new}} \leftarrow \text{top}_5(\mathcal{P})$
3. For remaining genomes $\mathcal{G}_{\text{remaining}}$:
 - (a) Calculate minimum distance to current population:

$$d_{\min}(\mathbf{g}) = \min_{\mathbf{g}' \in \mathcal{P}_{\text{new}}} d(\mathbf{g}, \mathbf{g}')$$

- (b) Select genome maximising diversity with performance constraint:

$$\mathbf{g}^* = \arg \max_{\mathbf{g} \in \mathcal{G}_{\text{remaining}}} d_{\min}(\mathbf{g}) \quad \text{s.t. } f(\mathbf{g}) \geq 0.7 \cdot f(\mathbf{g}_{\text{best}})$$

- (c) Add \mathbf{g}^* to \mathcal{P}_{new} and remove from $\mathcal{G}_{\text{remaining}}$

4. Repeat step 3 until the desired population size reached ($N + M$)

1458 H.3 GENETIC APPROACH: PSEUDO CODE

1459

1460

1461 **Algorithm 1** TAME: Genetic Algorithm1462 **Require:** Game environment1463 **Ensure:** Optimised genome g^*

```

1: Initialize  $P_0 = \{g_{\text{basic}}, g_{\text{hierarchical}}, g_{\text{default}}, g_{\text{full}}\}$ 
2: for each  $g \in P_0$  do
3:    $g_{\text{fitness}} \leftarrow \text{EvaluateFitness}(g)$ 
4: end for
5:  $P \leftarrow P_0$ 
6: for generation = 1 to GENERATIONS do
7:    $C \leftarrow \emptyset$  {Children population}
8:   for  $i = 1$  to CHILDREN_PER_GENERATION do
9:     if  $\text{rand}() < 0.7$  then
10:       $p_1 \leftarrow \text{RouletteWheelSelection}(P)$ 
11:       $c \leftarrow \text{SingleParentOperations}(p_1)$ 
12:    else
13:       $p_1, p_2 \leftarrow \text{RouletteWheelSelection}(P, 2)$ 
14:       $c \leftarrow \text{TwoParentOperations}(p_1, p_2)$ 
15:    end if
16:     $c_{\text{fitness}} \leftarrow \text{EvaluateFitness}(c)$ 
17:     $C \leftarrow C \cup \{c\}$ 
18:   end for
19:    $P \leftarrow \text{PopulationPruning}(P \cup C)$ 
20: end for
21:
22: return  $\arg \max_{g \in P} g_{\text{fitness}} = 0$ 

```

1485

1486

I MODULES ACTIVATED

1487

1488

In this Appendix we discuss module activation based on the game. The results are taken from the first one of genetic algorithm using Gemini-2.0-Flash. Module activation is based on final genomes returned by genetic algorithm, available in Appendix R. Module activation plot is demonstrated in Figure 15. When selecting the baseline configuration, no additional modules apart from Long-Term Memory can be activated. We notice that 4 out of 6 environments selected hierarchical module, highlighting the effectiveness of complex goal decomposition. TextWorld is a text-based environment where it is difficult to predict next actions due to their dependence on current observation, therefore hierarchical structure and memory are not adding value. Moreover, MiniHack has relatively short length (100 steps), which might be also why baseline structure was favoured.

1489

1490

1491

1492

1493

1494

1495

1496

1497

1498

1499

1500

1501

1502

1503

1504

1505

1506

1507

1508

1509

1510

1511

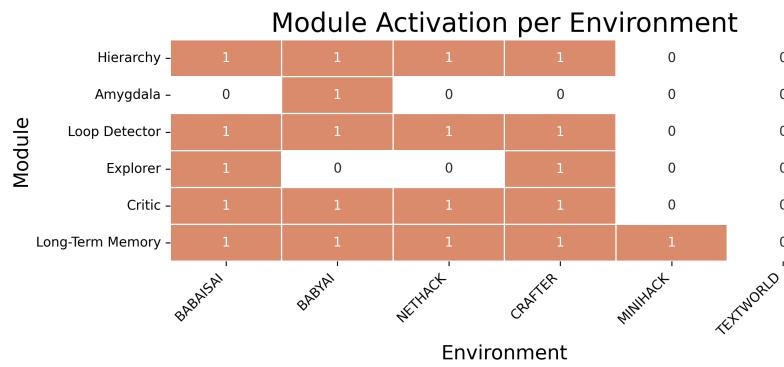
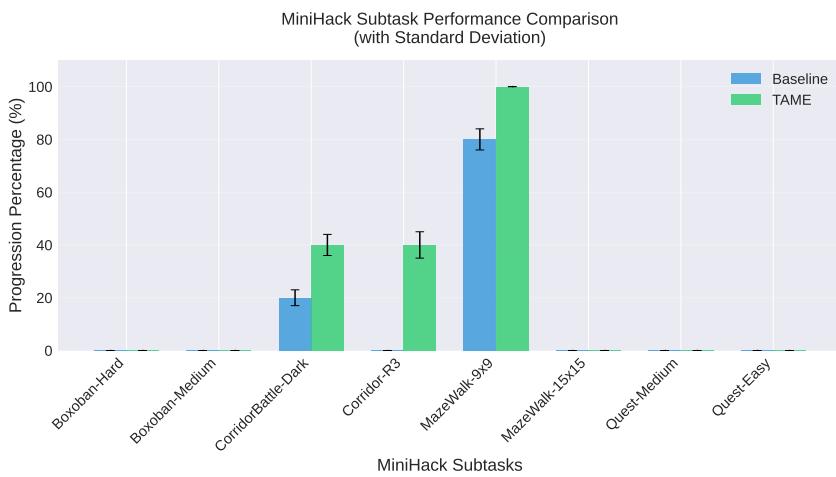


Figure 15: Module activation in TAME across environments.

1512 **J TASK PERFORMANCE**
1513

1514 In this Appendix, we compare different tasks performance across BabyAI, BabaIsAI, and Minihack
 1515 for baseline versus TAME (first run using Gemini-2.0-Flash). Crafter and Nethack are excluded
 1516 because they each have only one default task. TextWorld is also excluded since its genetic output
 1517 matches the baseline.

1518
 1519 **MiniHack** Interestingly, five out of eight tasks are never solved by any method, showcasing the
 1520 difficulty (see Figure 16). The Corridor-R3 task, which is never completed by the baseline, never-
 1521 theless shows 40% progress with TAME. Corridor-R3 is an exploration problem in which the goal
 1522 is to find the staircase Team (2024), illustrating TAME agent’s improved exploration ability. In both
 1523 CorridorBattle-Dark and MazeWalk-9×9, TAME achieves higher performance. CorridorBattle-Dark
 1524 requires the agent to fight monsters, thereby testing planning and memory MiniHack Team, whereas
 1525 MazeWalk-9×9 is a maze in which the agent must reach a terminal goal, testing exploration and
 1526 memory Samvelyan et al. (2021).
 1527

1544 Figure 16: MiniHack tasks progression, Baseline vs TAME using Gemini-2.0-Flash.
 1545
 1546
 1547

1548 **BabyAI** We observe clear performance improvements across all tasks except “putnext” where both
 1549 agents achieve 0% success rate (see Figure 17). The putnext task presents significant challenges
 1550 due to its complex spatial reasoning requirements. Through empirical analysis, we identified that
 1551 agents fail to understand the necessary positioning strategy: they must navigate to a location one
 1552 step away from the target position (which is adjacent to the object) before dropping the item. The
 1553 persistence of this failure in our improved method highlights fundamental limitations in the agent’s
 1554 spatial reasoning capabilities. In all the other tasks, we notice an improvement when comparing
 1555 TAME with baseline.

1556
 1557 **BabaIsAI** This environment consists of 40 distinct tasks that can be categorised into four main
 1558 types: make_win, make_you, goto_win, and make_wall_win. Our analysis reveals substantial im-
 1559 provements in the goto_win category and notable progress in make_win tasks, where performance
 1560 increased from 0% baseline (see Figure 18). When examining performance across different room
 1561 configurations (two_room versus single_room layouts), we observe consistent improvements in both
 1562 settings. For difficulty categorisation, we define three levels based on task complexity: simple
 1563 tasks have no modifiers or distractors, medium tasks contain 1-2 modifiers/distractors, and com-
 1564 plex tasks have more than 2 modifiers/distractors. Most notably, the greatest performance gains
 1565 occur in medium and complex categories, demonstrating that our method is particularly effective for
 challenging scenarios that require sophisticated reasoning capabilities.

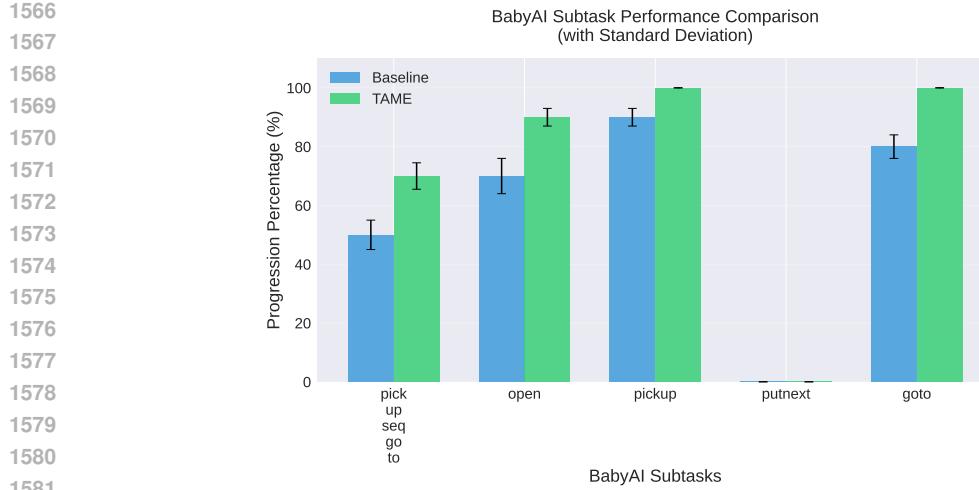


Figure 17: BabyAI tasks progression, Baseline vs TAME using Gemini-2.0-Flash.

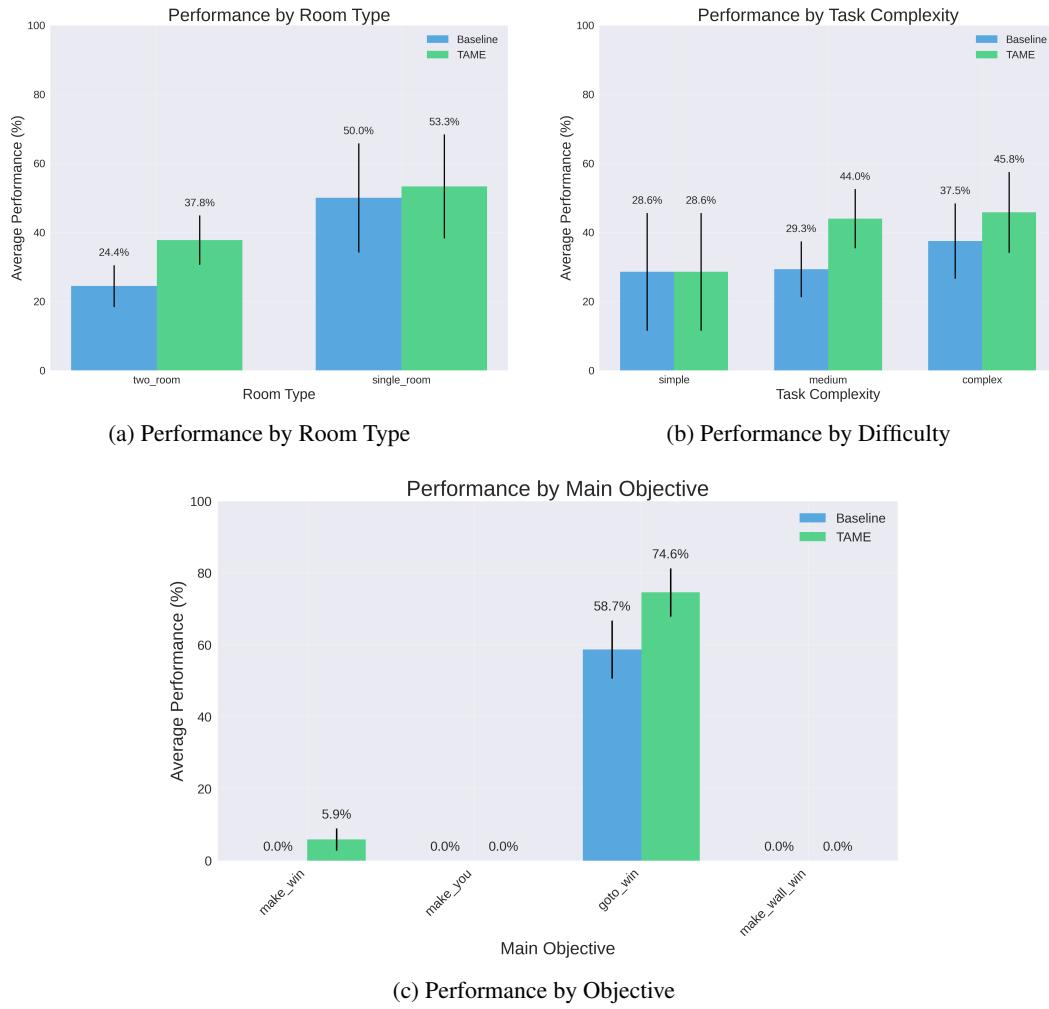


Figure 18: BabaIsAI task progression, Baseline vs TAME using Gemini-2.0-Flash

1620 K DETAILS OF GEMINI EXPERIMENTS

1621
 1622 This section details the setup and exact result of our Gemini model family. In our experimentation
 1623 we repeat the genetic algorithm 3 times in order to access its variance across runs. The number of
 1624 runs per episode remains the same as in BALROG benchmark.

1625 K.1 ‘IMPROVED PROMPT’ GEMINI RESULTS

1626 First, we compare Baseline together with “Improved Prompt”, and TAME. “Improved Prompt” (see
 1627 Appendix O) is created by prompting Claude-Sonnet-4.5 for a prompt that improves the Baseline
 1628 prompt and allowing it for functional inputs as described in 3.1 . This experiment demonstrates
 1629 the difficulty of crafting a prompt that works well for every environment, motivating our genetic
 1630 approach, adapting the framework based on the game.

1633 Environment	1634 Baseline (↑)	1635 ImprovedPrompt[Claude] (↑)	1636 TAME (↑)	1637 Episodes
1638 Average	1639 $27.16\% \pm 2.12\%$	1640 $23.36\% \pm 1.44\%$	1641 $35.05\% \pm 2.24\%$	-
babyai	$58.00\% \pm 6.98\%$	78.00% $\pm 5.86\%$	$72.00\% \pm 6.65\%$	50
babaisai	$30.83\% \pm 4.22\%$	$27.50\% \pm 4.08\%$	42.50% $\pm 4.51\%$	120
textworld	$32.55\% \pm 6.95\%$	$0.78\% \pm 0.77\%$	33.40% $\pm 7.23\%$	30
crafter	$29.09\% \pm 4.51\%$	$26.36\% \pm 2.47\%$	38.18% $\pm 4.25\%$	10
minihack	$12.50\% \pm 5.23\%$	$7.50\% \pm 4.16\%$	23.33% $\pm 6.69\%$	40
nle	$0.00\% \pm 0.00\%$	$0.00\% \pm 0.00\%$	0.91% $\pm 0.44\%$	5

1642 Table 6: “Improved Prompt” results with Gemini-2.0-flash
 1643
 1644

1645 K.2 SCORES ACROSS 3 GENETIC ALGORITHM RUNS WITH GEMINI-2.0-FLASH

1646 In Table 7 we see detailed scores across all environments and across all 3 independent genetic
 1647 algorithm runs. The main results in the Table 1 are averaged across those 3 runs.

1648 Environment	1649 Baseline (↑)	1650 TAME[Run1] (↑)	1651 TAME[Run2] (↑)	1652 TAME[Run3] (↑)
1653 Average	1654 $27.16\% \pm 2.12\%$	1655 $34.78\% \pm 2.22\%$	1656 $34.87\% \pm 2.20\%$	1657 $35.52\% \pm 2.24\%$
babyai	$58.00\% \pm 6.98\%$	$72.0\% \pm 6.35\%$	$68.0\% \pm 6.60\%$	$76.0\% \pm 6.04\%$
babaisai	$30.83\% \pm 4.22\%$	$41.67\% \pm 4.50\%$	$43.33\% \pm 4.52\%$	$42.5\% \pm 4.51\%$
textworld	$32.55\% \pm 6.95\%$	$32.55\% \pm 6.95\%$	$32.55\% \pm 6.95\%$	$35.10\% \pm 7.78\%$
crafter	$29.09\% \pm 4.51\%$	$39.09\% \pm 4.9\%$	$39.55\% \pm 3.91\%$	$35.91\% \pm 4.33\%$
minihack	$12.50\% \pm 5.23\%$	$22.5\% \pm 6.6\%$	$25.0\% \pm 6.85\%$	$22.50\% \pm 6.61\%$
nle	$0.00\% \pm 0.00\%$	$0.85\% \pm 0.47\%$	$0.79\% \pm 0.44\%$	$1.10\% \pm 0.41\%$

1658 Table 7: Detailed scores across 3 genetic algorithm runs with Gemini-2.0-Flash.
 1659
 1660

1661 K.3 INITIAL POPULATION SCORES

1662 This section details the scores of initial population used during genetic algorithm. Scores per environment can be seen in Table 8. The details of each components of initial population can be found in the Appendix Q.2.

1663 K.4 INITIAL POPULATION IMPROVEMENT

1664 In order to access the performance gain of genetic algorithm, we compare it against initial population.
 1665 In the Table 9 we present the percentage improvement of final TAME score when compared to the best score from initial population (see Appendix K.3). Those results show that even with
 1666 relatively small budget, TAME structure is able to improve on average of 25.27% when compared to initial population, showcasing the effectiveness of the genetic algorithm over human-crafted initial population.

Environment	Baseline (\uparrow)	Hierarchical[Hand-Crafted] (\uparrow)	Full[Claude] (\uparrow)	Full[Hand-Crafted] (\uparrow)
Average	$38.80\% \pm 5.32\%$	$38.48\% \pm 4.06\%$	$34.53\% \pm 3.99\%$	$33.93\% \pm 4.07\%$
babyai	$58.00\% \pm 6.98\%$	$62.00\% \pm 6.86\%$	$58.00\% \pm 6.98\%$	$65.31\% \pm 6.80\%$
babaisai	$30.83\% \pm 4.22\%$	$29.17\% \pm 4.15\%$	$40.83\% \pm 4.49\%$	$31.67\% \pm 4.25\%$
textworld	$32.55\% \pm 6.95\%$	$8.24\% \pm 2.25\%$	$4.67\% \pm 1.50\%$	$6.08\% \pm 1.70\%$
crafter	$29.09\% \pm 4.51$	$33.64\% \pm 4.64\%$	$31.36\% \pm 3.55\%$	$20.00\% \pm 1.72\%$
minihack	$12.50\% \pm 5.23\%$	$5.00\% \pm 3.45\%$	$12.50\% \pm 5.23\%$	$12.50\% \pm 5.23\%$
nle	$0.00\% \pm 0.00\%$	$0.37\% \pm 0.33\%$	$0.68\% \pm 0.37\%$	$0.00\% \pm 0.00\%$

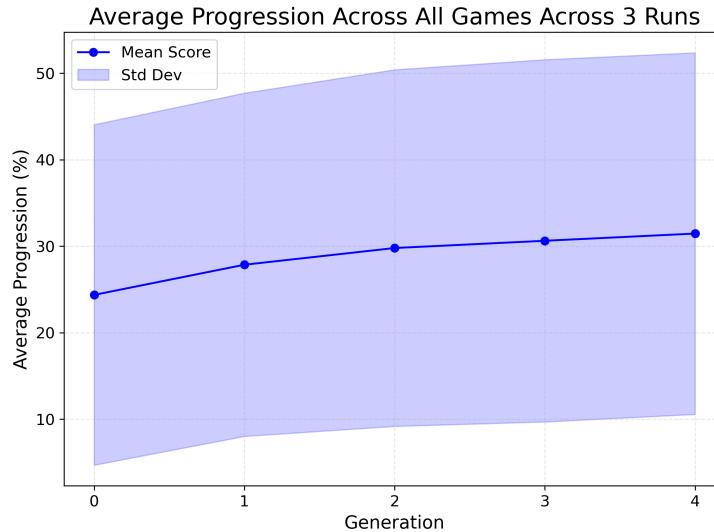
Table 8: Initial population for Gemini-2.0-Flash genetic algorithm.

Environment	Relative Improvement (\uparrow)
Average	$25.27\% \pm 3.05\%$
TAME[Run 1]	22.24%
TAME[Run 2]	24.00%
TAME[Run 3]	29.47%

Table 9: Relative percentage improvement of TAME over the best-performing initial population scores across environments using Gemini-2.0-Flash. *Note: The values indicate the percentage progression relative to the best performing initial genome, not the raw scores themselves.*

K.5 CONVERGENCE PROPERTIES OF GENETIC ALGORITHM

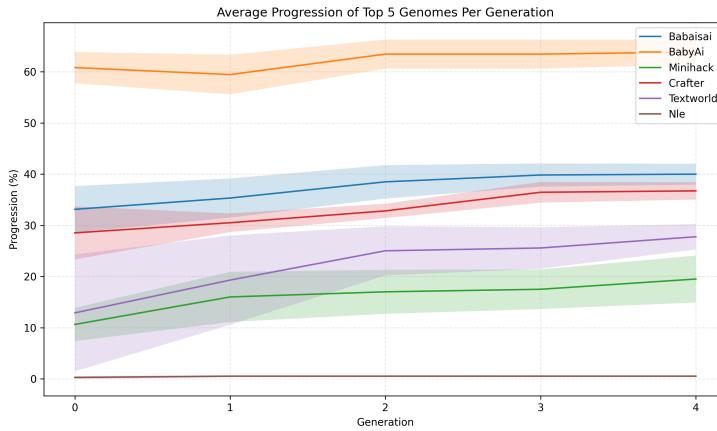
We also show in Figure 19 the graphical representation of average game progression (% completion) over generations in genetic algorithm with Gemini-2.0-Flash. In each generation we consider 5 top scoring genomes. We present average and standard deviation of the score across all 6 games over 3 runs, showcasing that there is a clear improvement in terms of average progress, which shows promise for convergence when algorithm is run for longer.

Figure 19: Average progression across all games (3 runs) with \pm standard deviation. *Note: Values are absolute scores, not relative improvements.*

K.5.1 CONVERGENCE PROPERTIES ACROSS GAMES

Figure 20 expands on Figure 19, presenting the convergence patterns across all games. We measure the mean and standard deviation of the top 5 scoring genomes in the population at each generation. The results show a clear convergence trend: the standard deviation decreases in all environments

1728 except for MiniHack. This demonstrates that increasing the computational budget enables convergence.
 1729 Note that due to the scale of the y-axis, NetHack’s standard deviation appears negligible in
 1730 the figure. However, Table 1 confirms that measurable variance exists and that TAME continues to
 1731 show improvement in this environment.
 1732



1733 Figure 20: Average progression of top 5 scoring genomes in each generation over environments
 1734 using Gemini-2.0-Flash. *Note: Values are absolute scores, not relative improvements.*
 1735

1736 K.6 GENOME TRANSFERABILITY TO OTHER GEMINI MODELS

1737 In this Appendix we detail transerability of genomes obtained through genetic algorithm using
 1738 Gemini-2.0-Flash, to Gemini-2.5-Flash-Lite and Gemini-2.5-Pro without additional training. In the
 1739 Table 10 we compare performance of Baseline using Gemini-2.5-Flash-Lite vs TAME transferred
 1740 to the same model. In the Table 11 we compare performance of Baseline using Gemini-2.5-Pro
 1741 vs TAME transferred to the same model. Lastly, we show combined results in the Figure 21. The
 1742 Transferred results are averaged over 3 independent runs of genetic algorithm.
 1743

Environment	Baseline [2.5-Flash-Lite]	TAME[Transferred]	Episodes
Average	11.87% \pm 1.32%	20.48% \pm 0.91%	-
babyai	46.00% \pm 7.05%	58.00% \pm 3.06%	50
babai	9.17% \pm 2.63	38.33% \pm 0.48%	120
textworld	7.45% \pm 2.30%	7.45% \pm 0.00%	30
crafter	8.64% \pm 1.00%	13.01% \pm 4.15%	10
minihack	0.00% \pm 0.00%	5.83% \pm 1.67%	40
nle	0.00% \pm 0.00%	0.27% \pm 0.15%	5

1750 Table 10: Comparison of Baseline vs. TAME[Transferred] using Gemini-2.5-Flash-Lite.
 1751 TAME[Transferred] averaged across 3 genetic algorithm runs.
 1752

1753 K.6.1 TRANSFER SCORES ACROSS INDIVIDUAL GENETIC RUNS

1754 In the Table 12 we present detailed results of genetic algorithm transferability to Gemini-2.5-Pro.
 1755 Similarly, in Table 13 we show results of genetic algorithm transferability to Gemini-2.5-Flash-Lite
 1756 across 3 independent genetic runs.
 1757

1758 L DETAILS OF GPT EXPERIMENTS

1759 In this section we aim to test if we can get similar improvement when applying TAME to another
 1760 family of Language Models. We test GPT4.1-nano, as it provides a fast inference, and we also test
 1761 transferability to GPT4.1-mini. We run genetic algorithm 3 times and average results.
 1762

Game	Baseline [2.5-Pro]	Top Model	TAME[Transferred]	Episodes
Average	43.35% \pm 2.31%	43.60% \pm 2.17%	47.57% \pm 2.72%	-
babyai	80.0% \pm 5.70%	76.00% \pm 6.00%	92.00% \pm 3.46%	50
babaisai	56.70% \pm 4.50%	45.80% \pm 4.50%	67.50% \pm 5.00%	120
textworld	49.20% \pm 8.20%	62.90% \pm 7.90%	52.73% \pm 6.12%	30
crafter	55.0% \pm 6.0%	57.30% \pm 3.90%	55.00% \pm 0.00%	10
minihack	17.50% \pm 6.00%	17.5% \pm 6.00%	18.33% \pm 1.44%	40
nle	1.70% \pm 0.20%	1.80% \pm 0.8%	1.70% \pm 0.00%	5

Table 11: Comparison of Baseline vs. TAME[Transferred] using Gemini-2.5-Pro. TAME[Transferred] averaged across 3 genetic algorithm runs.

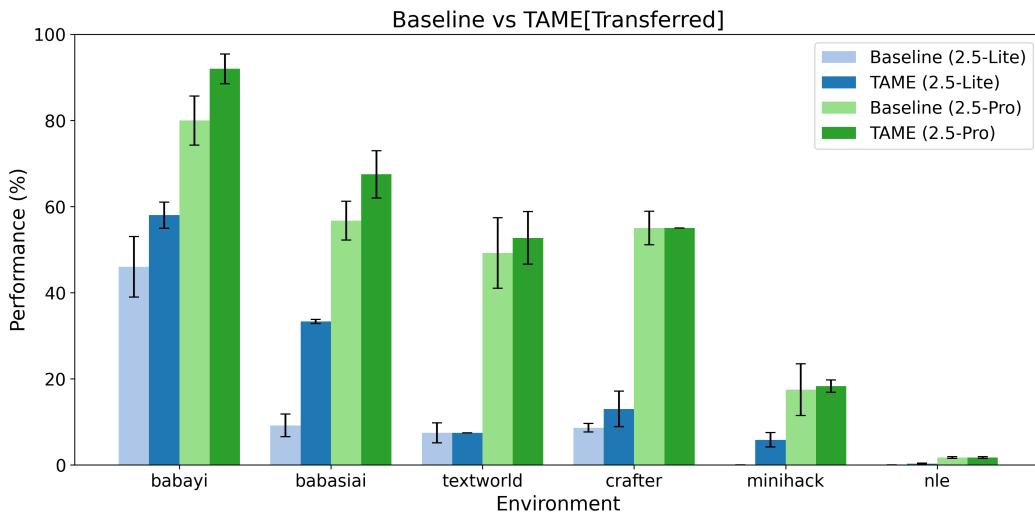


Figure 21: Evaluation of TAME Structure Transferability: Identical performance to the baseline indicates that the genetic algorithm favored the baseline architecture over the transferred TAME genome.

L.1 TAME RESULTS OVER GPT MODEL FAMILY

We repeat the same procedure as described in section 3.2 applied to the initial population described in Section L.2. Table 14 shows the final TAME result, illustrating the relative gain of 74% when comparing average result over Baseline (see Table 15 for detailed results from independent runs). Additionally, we include results of applying the “Improved Prompt” proposed by Claude-Sonnet-4.5 (see Appendix O). We conclude that TAME outperforms both baseline and ImprovedPrompt results. Those results demonstrate that TAME architecture are transferrable on more than onefamily of Language Models.

L.2 INITIAL POPULATION OF GPT FAMILY MODELS

Table 16 shows initial population for GPT4.1-nano genetic algorithm. It follows the same structure as Gemini experiment and each component is described more in detail in the Appendix Q.2.

L.3 IMPROVEMENT OVER INITIAL POPULATION

Table 17 shows the relative improvement when TAME is contrasted with best scoring genome from initial population described in Appendix L.2. The results demonstrate average relative gain of around 37.10%, showing the effectiveness of genetic algorithm over human-crafted initial population. Note that since NetHack baseline performance is 0%, even though we notice an improvement

Environment	Baseline[2.5-Pro]	TAME[Run1]	TAME[Run2]	TAME[Run3]	Episodes
Average	$43.35\% \pm 2.31\%$	$47.65\% \pm 2.20\%$	$47.82\% \pm 2.14\%$	$48.17\% \pm 2.25\%$	
babyai	$80.00 \pm 5.70\%$	90.00 ± 4.24	$96.0 \pm 2.77\%$	$90\% \pm 4.24\%$	50
babaisai	$56.70\% \pm 4.50\%$	$72.50\% \pm 4.08\%$	$67.5\% \pm 4.28\%$	$62.5\% \pm 4.42\%$	120
textworld	$49.20\% \pm 8.20\%$	$49.20\% \pm 8.20\%$	$49.20\% \pm 8.20\%$	$59.80\% \pm 8.34\%$	30
crafter	$55.0\% \pm 6.00\%$	$55.00\% \pm 6.00\%$	$55.00\% \pm 6.00\%$	$55.00\% \pm 6.00\%$	10
minihack	$17.50\% \pm 6.00\%$	$17.50\% \pm 6.00\%$	$17.50\% \pm 6.00\%$	$20.00\% \pm 6.32\%$	40
nle	$1.70\% \pm 0.20\%$	$1.70\% \pm 0.20\%$	$1.70\% \pm 0.20\%$	$1.70\% \pm 0.20\%$	5

Table 12: Details of independent runs of transferability to Gemini-2.5-Pro

Environment	Baseline[2.5-Flash-Lite]	TAME[Run1]	TAME[Run2]	TAME[Run3]	Episodes
Average	$11.88\% \pm 1.32\%$	$19.25\% \pm 1.63\%$	$19.43\% \pm 1.49\%$	$22.76\% \pm 1.73\%$	
babyai	$46.00\% \pm 7.05\%$	$52.00\% \pm 7.07\%$	$60.00\% \pm 6.93\%$	$62.00\% \pm 6.93\%$	50
babaisai	$9.17\% \pm 2.63$	$39.17\% \pm 4.46\%$	$37.50\% \pm 4.42\%$	$38.33\% \pm 4.44\%$	120
textworld	$7.45\% \pm 2.30\%$	$7.45\% \pm 2.30\%$	$7.45\% \pm 2.30\%$	$7.45\% \pm 2.30\%$	30
crafter	$8.64\% \pm 1.00\%$	$9.09\% \pm 1.70\%$	$8.64\% \pm 1.00\%$	$21.3\% \pm 4.22\%$	10
minihack	$0.00\% \pm 0.00\%$	$7.5\% \pm 4.16\%$	$2.5\% \pm 2.47$	$7.50\% \pm 4.16\%$	40
nle	$0.00\% \pm 0.00\%$	$0.31\% \pm 0.28\%$	$0.51\% \pm 0.42\%$	$0.00\% \pm 0.00\%$	5

Table 13: Details of independent runs of transferability to Gemini-2.5-Flash-Lite

Environment	Baseline (\uparrow)	ImprovedPrompt[Claude] (\uparrow)	TAME (\uparrow)	Episodes
Average	$9.90\% \pm 1.33\%$	$6.10\% \pm 1.12\%$	$17.20\% \pm 1.47\%$	-
babyai	$32.0\% \pm 6.60\%$	$20.00\% \pm 5.66\%$	$48.67\% \pm 7.07\%$	50
babaisai	$12.5\% \pm 3.02\%$	$6.67\% \pm 2.28\%$	$21.55\% \pm 3.76\%$	120
textworld	$0.59\% \pm 0.58\%$	$0.39\% \pm 0.39\%$	$2.88\% \pm 0.94\%$	30
crafter	$11.82\% \pm 2.15\%$	$7.07\% \pm 1.26\%$	$19.78\% \pm 2.33\%$	10
minihack	$2.5\% \pm 2.47\%$	$2.50\% \pm 2.47\%$	$10.00\% \pm 2.78\%$	40
nle	$0.0\% \pm 0.0\%$	$0.00\% \pm 0.00\%$	$0.22\% \pm 0.20\%$	5

Table 14: Baseline vs. ImprovedPrompt[Claude] vs. TAME using GPT4.1-nano averaged over 3 runs.

after applying TAME, we can't measure its relative improvement, therefore we exclude it from calculation in Table 17.

L.4 TRANSFERABILITY ACROSS GPT FAMILY MODELS

In this section we test if results can transfer within GPT family. We follow the same procedure as in Gemini case. Therefore, for each environment we evaluate the best scoring genome (obtained from TAME run with GPT4.1-nano) using GPT4.1-mini. We apply the same procedure for all independent genetic algorithm runs and average results. Table 18 shows the results, demonstrating a relative gain of around 10%, when compared to Baseline GPT4.1-mini results (see Table 19 for more details across independent runs).

M TAME CHILD EVALUATION PSEUDO CODE

This Appendix presents the pseudo code behind TAME evaluation on specific game when all modules are active, as decided by the genetic algorithm (overview in Figure 2). The algorithm works as follows: until maximum number of steps is reached, with probability ϵ choose exploration, otherwise exploitation. For each option, it retrieves top successful and failed memories, then the Low-level Executor decides and executes actions. If danger is detected, the survival module is triggered and execution stops. Otherwise, the Critic evaluates success, summarizes key actions, and in case of loops, the summary is replaced by the Loop Detector summary. The resulting memory is stored, and the cycle repeats.

Environment	Baseline (\uparrow)	TAME[Run1] (\uparrow)	TAME[Run2] (\uparrow)	TAME[Run3] (\uparrow)
Average	$9.90\% \pm 1.33\%$	$18.06\% \pm 1.65\%$	$17.19\% \pm 1.60\%$	$16.31\% \pm 1.58\%$
babyai	$32.00\% \pm 6.60\%$	$52.00\% \pm 7.07\%$	$48.00\% \pm 7.07\%$	$46.00\% \pm 7.06\%$
babaisai	$12.50\% \pm 3.02\%$	$22.5\% \pm 3.81\%$	$22.5\% \pm 3.81\%$	$19.66\% \pm 3.67\%$
textworld	$0.59\% \pm 0.58\%$	$3.13\% \pm 0.95\%$	$2.75\% \pm 0.93\%$	$2.75\% \pm 0.93\%$
crafter	$11.82\% \pm 2.15\%$	$20.71\% \pm 3.04\%$	$19.55\% \pm 2.14\%$	$19.09\% \pm 1.80\%$
minihack	$2.5\% \pm 2.47\%$	$10.00\% \pm 4.86\%$	$10.00\% \pm 4.74\%$	$10.00\% \pm 4.74\%$
nle	$0.0\% \pm 0.0\%$	$0.0\% \pm 0.0\%$	$0.31\% \pm 0.28\%$	$0.38\% \pm 0.33\%$

Table 15: Detailed scores across 3 genetic algorithm runs with GPT4.1-nano.

Environment	Baseline (\uparrow)	Hierarchical[Hand-Crafted] (\uparrow)	Full[Claude] (\uparrow)	Full[Hand-Crafted] (\uparrow)
Average	$9.90\% \pm 1.33\%$	$10.40\% \pm 2.07\%$	$10.95\% \pm 2.77\%$	$9.28\% \pm 2.15\%$
babyai	$32.0\% \pm 6.60\%$	$12.00\% \pm 4.60\%$	$22.00\% \pm 5.86\%$	$10.00\% \pm 4.24\%$
babaisai	$12.50\% \pm 3.02\%$	$16.67\% \pm 3.40\%$	$16.10\% \pm 3.38\%$	$19.66\% \pm 3.67\%$
textworld	$0.59\% \pm 0.58\%$	$1.18\% \pm 0.60\%$	$2.75\% \pm 0.93\%$	$1.37\% \pm 0.70\%$
crafter	$11.82\% \pm 2.15\%$	$7.58\% \pm 1.01\%$	$15.00\% \pm 1.71\%$	$12.27\% \pm 1.82\%$
minihack	$2.50\% \pm 2.47\%$	$5.00\% \pm 3.45\%$	$0.00\% \pm 0.00\%$	$2.50\% \pm 2.47\%$
nle	$0.00\% \pm 0.00\%$	$0.00\% \pm 0.00\%$	$0.00\% \pm 0.00\%$	$0.00\% \pm 0.00\%$

Table 16: Initial population for GPT-4.1-nano genetic algorithm.

N DETAILED MEMORY ABLATIONS

In this section we show detailed results when comparing our memory system with Craft-Jarvis-1 and A-Mem using Gemini-2.0-Flash. Table 20 shows that TAME-Memory consistently outperforms both Craft-Jarvis-1 and A-Mem on most environments, yielding the highest overall average of 23.11%. The gains are particularly strong in TextWorld and Crafter, where our system nearly doubles or surpasses the baselines by a large margin. While performance is comparable in Minihack and NLE, these tasks remain challenging for all methods. Overall, the results highlight the efficiency of our hybrid memory system.

O ‘IMPROVED PROMPT’ PROPOSED BY CLAUDE

Here we present a prompt proposed by Claude-Sonnet-4.5, when tasked to improve the Baseline (BALROG) prompt. As well as the baseline prompt, we gave information about functional inputs (see Section 3.1) and gave it an option to use them.

Improved Prompt proposed by Claude

```

# Game Agent Instructions

You are an autonomous agent navigating a game environment. Your task is to select the optimal next
    ↢ action based on the current state.

## Current State
- **Observation**: {obs}
- **Current Subgoal**: {subgoal}
- **Success Condition**: {success_condition}

## Context
- **Recent History**: {history}
- **Action Sequence (Current Subgoal)**: {action_sequence}

## Knowledge Base
- **Similar Successful Attempts**: {entries_successful_subgoal}
- **Similar Failed Attempts**: {entries_failed_subgoal}

## Available Actions
{game_info}

## Output Format

```

Environment	Relative Improvement (↑)
Average	37.10% \pm 2.08%
TAME[Run 1]	38.14%
TAME[Run 2]	38.96%
TAME[Run 3]	34.20%

Table 17: Relative percentage improvement of TAME over the best-performing initial population scores across environments using GPT4.1-nano averaged over 3 genetic algorithm runs. *Note: The values indicate the percentage progression relative to the best performing initial genome, not the raw scores themselves.*

Environment	Baseline[GPT-4.1-mini]	TAME[Transferred]	Episodes
Average	24.43% \pm 1.89%	26.80% \pm 1.92%	-
babyai	72.00% \pm 7.18%	72.00% \pm 7.18%	50
babaisai	29.17% \pm 4.15%	35.00% \pm 4.33%	120
textworld	13.33% \pm 5.74%	13.33% \pm 5.74%	30
crafter	21.00% \pm 2.00%	29.43% \pm 2.65%	10
minihack	10.00% \pm 4.74%	10.00% \pm 4.74%	40
nle	1.09% \pm 0.41%	1.09% \pm 0.41%	5

Table 18: Transferability of TAME across GPT family averaged over 3 runs.

```
Output ONLY a single valid action from the available actions list. No explanations, no additional
→ text.

## Decision Strategy
1. Analyze current observation against the success condition
2. Learn from similar past attempts (successful and failed)
3. Consider recent action history to avoid loops
4. Select the action most likely to progress toward the subgoal
Output in the following format: \n
    REASONING: <your reasoning> \n
    ACTION: <your action>
```

P LLM MUTATION AND CROSSOVER PROMPTS

This Appendix details the prompts used for LLM based Crossover (in the case of single parent) and LLM based Crossover and Mutation (in the case of two parents) used in genetic algorithm. Prompts created with the help of Claude-Sonnet-4.

LLM Mutation Prompt

```
"""
Please follow the instruction step-by-step to generate a better prompt.

1. Consider prompt:
Prompt 1: <prompt1>

2. Apply ONE of the following mutation strategies to create a significantly different prompt:

**Strategy A - Perspective Shift:** Change the role/perspective (e.g., "As an expert analyst..." or
    ↪ "From the viewpoint of...")

**Strategy B - Methodology Change:** Alter the approach (step-by-step -> holistic analysis, direct
    ↪ -> comparative, etc.)

**Strategy C - Output Format Transformation:** Change how results are presented (narrative ->
    ↪ structured, single response -> multi-part, etc.)

**Strategy D - Contextual Enhancement:** Add specific domain knowledge or constraints that weren't
    ↪ in the original
```

Environment	Baseline[GPT4.1-mini]	TAME[Run1]	TAME[Run2]	TAME[Run3]	Episodes
Average	$24.43\% \pm 4.04\%$	$28.33\% \pm 4.17\%$	$25.40\% \pm 1.09\%$	$25.44\% \pm 1.92\%$	-
babyai	$72.00\% \pm 7.18\%$	$72.00\% \pm 7.18\%$	$72.00\% \pm 7.18\%$	$72.00\% \pm 7.18\%$	50
babaiasai	$29.17\% \pm 4.15\%$	$40.83\% \pm 4.49\%$	$35.00\% \pm 4.35\%$	$29.17\% \pm 4.15\%$	120
textworld	$13.33\% \pm 5.74\%$	$13.33\% \pm 5.74\%$	$13.33\% \pm 5.74\%$	$13.33\% \pm 5.74\%$	30
crafter	$21.00\% \pm 2.00\%$	$32.73\% \pm 2.47\%$	$21.00\% \pm 2.00\%$	$34.55\% \pm 3.47\%$	10
minihack	$10.00\% \pm 4.74\%$	$10.00\% \pm 4.74\%$	$10.00\% \pm 4.74\%$	$10.00\% \pm 4.74\%$	40
nle	$1.09\% \pm 0.41\%$	$1.09\% \pm 0.41\%$	$1.09\% \pm 0.41\%$	$1.09\% \pm 0.41\%$	5

Table 19: Details of independent runs of transferability to GPT4.1-mini

Algorithm 2 TAME [Full Structure]

Require: I, s_0
Ensure: st (final game state)

while $num_steps \leq max_steps$ **do**

if $rand() < \epsilon$ **then**
 $g \leftarrow \pi_{explorer}(s_i^e)$ {Exploration}
 else
 $g \leftarrow \pi_{high}(s_i^{mc})$ {Exploitation}
 end if

for $g_i \in g$ **do**
 $M_i^+ \leftarrow retrieve_successful(n_{g_i}, d_{g_i}, memory)$
 $M_i^- \leftarrow retrieve_failed(n_{g_i}, d_{g_i}, memory)$
 $a \leftarrow \pi_{low}(s_i^{le})$
 for $a_i \in a$ **do**
 $s_t \leftarrow execute_action(a, s_i)$
 $s_i \leftarrow s_t$
 if $\sigma(s_i)$ **then**
 $g \leftarrow g^*$ {Activate survival option}
 break {Move to option selection and force g^* }
 end if
 end for
 $C_i \leftarrow \varphi(s_i)$
 $h_i \leftarrow \rho(s_i, C_i)$
 if C_i **then**
 $store_memory(g_i, h_i, o, "successful")$
 else
 if $\psi(s_i)$ **then**
 $h_i \leftarrow \psi(s_i)$
 end if
 $store_memory(g_i, h_i, o, "failed")$
 break {Escalate to replanning}
 end if
 end for
end while=0

2038

2039 ****Strategy E - Complexity Modulation:**** Significantly increase or decrease the cognitive complexity
2040 → of the task

2041 ****Strategy F - Functional Input Integration:**** Incorporate functional inputs in a novel way that
2042 → changes the prompt's core operation

2043 3. Generate the final mutated prompt bracketed with <prompt> and </prompt>.

2044 ## Available Functional Inputs:
2045 - {obs} : current observation
2046 - {game_info} : information about the game (possible actions and goal)
2047 - {subgoal} : current subgoal that you're working towards
2048 - {action_sequence}: action sequence towards current subgoal
2049 - {observation_sequence}: observation sequence towards current subgoal
2050 - {success_condition}: termination condition of the current subgoal
2051 - {action_obs_seq}: action-observation pairs towards current subgoal
2052 - {survival_plan}: survival plan
2053 - {history}: history of the last 16 action-observation pairs
2054 - {entries_successful_goal}: most similar successful subgoals to the current one
2055 - {entries_failed_goal}: most similar failed subgoals to the current one

Environment	Jarvis (\uparrow)	TAME-Memory[ours] (\uparrow)	A-mem (\uparrow)	Episodes
Average	$17.52\% \pm 1.73\%$	$23.11\% \pm 1.75\%$	$21.45\% \pm 1.80\%$	-
babyai	$48.00\% \pm 7.06\%$	$62.00\% \pm 6.86\%$	$58.00\% \pm 6.98\%$	50
babaisai	$24.17\% \pm 3.90\%$	$29.17\% \pm 4.15\%$	$26.67\% \pm 4.04\%$	120
textworld	$0.59\% \pm 0.58\%$	$8.24\% \pm 1.95\%$	$4.51\% \pm 2.35\%$	30
crafter	$19.55\% \pm 3.81\%$	$35.45\% \pm 3.20\%$	$26.36\% \pm 4.25\%$	10
minihack	$12.5\% \pm 5.23\%$	$3.45\% \pm 5.65\%$	$12.5\% \pm 5.23\%$	40
nle	$0.31\% \pm 0.28\%$	$0.37\% \pm 0.33\%$	$0.68\% \pm 0.37\%$	5

Table 20: Comparison of TAME average game progression across different memory types using Gemini-2.0-Flash.

LLM Mutation and Crossover Prompt

```

"""
Please follow the instruction step-by-step to generate a better prompt.

1. Crossover the following prompts and generate a new prompt:
Prompt 1: <prompt1>
Prompt 2: <prompt2>

2. Apply ONE of the following mutation strategies to create a significantly different prompt:
**Strategy A - Perspective Shift:** Change the role/perspective (e.g., "As an expert analyst..." or
    ↪ "From the viewpoint of...")
**Strategy B - Methodology Change:** Alter the approach (step-by-step → holistic analysis, direct
    ↪ → comparative, etc.)
**Strategy C - Output Format Transformation:** Change how results are presented (narrative →
    ↪ structured, single response → multi-part, etc.)
**Strategy D - Contextual Enhancement:** Add specific domain knowledge or constraints that weren't
    ↪ in the original
**Strategy E - Complexity Modulation:** Significantly increase or decrease the cognitive complexity
    ↪ of the task
**Strategy F - Functional Input Integration:** Incorporate functional inputs in a novel way that
    ↪ changes the prompt's core operation

3. Generate the final mutated prompt bracketed with <prompt> and </prompt>.

## Available Functional Inputs:
- {obs} : current observation
- {game_info} : information about the game (possible actions and goal)
- {subgoal} : current subgoal that you're working towards
- {action_sequence}: action sequence towards current subgoal
- {observation_sequence}: observation sequence towards current subgoal
- {success_condition}: termination condition of the current subgoal
- {action_obs_seq}: action-observation pairs towards current subgoal
- {survival_plan}: survival plan
- {history}: history of the last 16 action-observation pairs
- {entries_successful_goal}: most similar successful subgoals to the current one
- {entries_failed_goal}: most similar failed subgoals to the current one

**CRITICAL:** Use functional inputs with exact bracket names. Ensure the mutation creates a
    ↪ substantially different prompt that would produce notably different outputs than the
    ↪ crossover result.
Output your answer in the following way:
REASONING: <your reasoning>
PROMPT: <mutated prompt>
"""

```

2106 **Q INITIAL POPULATION GENOMES**

2108 This Appendix details the initial population of genetic algorithm described in section 3.2. First, we
 2109 show examples of prompts proposed by Claude-Sonnet-4 for each of the modules. We use those in
 2110 order to construct Full[Claude] genome. For the rest of genomes we use hand-crafted prompts. We
 2111 detail genomes in the subsections below.

2112 **Q.1 PROMPTS PROPOSED BY SONNET-4**

2113 Those prompts were created using Claude-Sonnet-4, which was tasked with the creation of a prompt
 2114 for each module. We call those “default” prompts

2115 **Prompt Template**

```

2116
2117
2118
2119
2120     default_sequential_prompt = """You always have to output one of the above actions at a time and no
2121     ↪ other text. You always have to output an action until the episode terminates."""
2122
2123     default_highlevel_prompt = """
2124     You are a strategic planner for a video game AI. Analyze the current game state and create
2125     ↪ achievable subgoals that advance toward the main objective.
2126
2127     REQUIREMENTS:
2128     - Subgoals must be immediately achievable with current capabilities
2129     - Focus on next logical steps, not distant goals
2130     - Each subgoal should have clear success criteria
2131
2132     CURRENT STATE: {obs}
2133     GAME INFO: {game_info}
2134
2135     Create a sequential plan with 2-3 subgoals."""
2136     default_lowlevel_prompt = """ You are an action executor in a video game AI system. Given a subgoal
2137     ↪ from the high-level planner, propose a sequence of actions to achieve it.
2138     ##########
2139     CURRENT SUBGOAL: {subgoal}
2140     CURRENT STATE: {obs}
2141     PREVIOUS ACTIONS: {action_sequence}
2142
2143     Plan the full sequence of actions needed to complete the subgoal. Avoid repeating actions if
2144     ↪ observations don't change.
2145     Avoid extra commentary outside the REASONING and ACTIONS list.
2146     """
2147
2148     default_termination_prompt = """
2149     You are a termination evaluator for a video game AI. Check if the agent has completed
2150     ↪ its subgoal.
2151
2152     CURRENT STATE: {obs}
2153     SUBGOAL: {subgoal}
2154     SUCCESS CONDITION: {success_condition}
2155     RECENT ACTIONS: {action_sequence}
2156
2157     Compare the current state with the success condition to determine if the subgoal is
2158     ↪ complete. Provide feedback to help the agent improve.
2159     """
2160
2161     default_summariser_prompt = """
2162     You are a critic analyzing an agent's subgoal attempt. Identify the key factor that caused
2163     ↪ success or failure.
2164
2165     SUBGOAL: {subgoal}
2166     OUTCOME: {outcome}
2167     ACTION HISTORY: {action_obs_seq}
2168
2169     Focus on specific resources and quantities that mattered most. If no resources involved,
2170     ↪ identify the next most important factor.
2171     """
2172
2173     default_amygdala_prompt = """
2174     Decide if survival mode should activate.
2175
2176     Observation: {obs}
2177     Survival plan: {survival_plan}
2178
2179     1. Check if observation meets any subtask prerequisites.
2180     2. If several match, pick highest priority.
2181     """
2182
2183     default_loop_prompt = """
2184     Task: Decide if the agent is stuck in a loop.
2185
2186     Loop = repeating actions without meaningful progress toward the subgoal
2187     (progress = closer to goal, new info, removing failed paths, or advancing game state).
2188
2189     Data:
2190     - Observation: {obs}
  
```

```

2160
2161     - Subgoal: {subgoal}
2162     - Termination condition: {success_condition}
2163     - Action-observation history: {history}
2164
2165     Steps:
2166     1. Check if enough steps have been taken to allow exploration.
2167     2. Look for repeated patterns without progress.
2168     3. If loop detected, identify cause: missing info, unknown prerequisite, or unexplored path.
2169     """
2170     default_explorer_prompt = """
2171         Task: Create an exploration plan to help the agent discover new skills.
2172
2173         Data:
2174         - Game info: {game_info}
2175         - Observation: {obs}
2176         - Subgoal summary: {summary}
2177         - Recent 16 action-observation pairs: {history}
2178
2179         Steps:
2180         1. Analyze the environment and agent's situation.
2181         2. Propose a focused exploration plan with clear purpose, conditions, and indicators.
2182
2183     """

```

Q.2 INITIAL POPULATION

Detailed genome descriptions for the initial population.

Baseline genome This genome corresponds to the baseline structure used in BALROG.

Baseline Genome

```

2184     genome_basic = {
2185         "id": "3",
2186         "hierarchy": 0,
2187         "amygdala": 0,
2188         "loop_evaluator": 0,
2189         "explorer": 0,
2190         "summariser": 0,
2191         "time_decay": [0, 0.01],
2192         "cosine_cutoff": [0, 0.05],
2193         "epsilon": [0, 0.01],
2194         "temperature": 1.0,
2195         "prompts": {
2196             "high_level": [0, default_highlevel_prompt],
2197             "low_level": [0, default_lowlevel_prompt],
2198             "amygdala": [0, default_amygdala_prompt],
2199             "loop": [0, default_loop_prompt],
2200             "termination": [0, default_termination_prompt],
2201             "summariser": [0, default_summariser_prompt],
2202             "explorer": [0, default_explorer_prompt],
2203             "sequential": [1, default_sequential_prompt]
2204         },
2205         "fitness": ...
2206     }

```

Hierarchical[Hand-Crafted] genome This genome corresponds to hierarchy from Section 3.3.1 combined with memory 3.4, without amygdala and explorer. We use this architecture as one of the initial seeds, because we found that it gives the most stable performance across all environments.

Hierarchical[Hand-Crafted] Genome

```

2206     genome_stable = {
2207         "id": "2",
2208         "hierarchy": 1,
2209         "amygdala": 0,
2210         "loop_evaluator": 1,
2211         "explorer": 0,
2212         "summariser": 1,
2213         "time_decay": [1, 0.01],
2214         "cosine_cutoff": [0, 0.1],
2215         "epsilon": [0, 0.01],
2216         "temperature": 1.0,
2217         "prompts": {
2218             "high_level": [1,
2219             ...
2220         }

```

```

2214
2215     """ You are a strategic planner in a hierarchical video game AI system. Your role is to analyze
2216     ↪ the current game state and generate achievable subgoals that strategically advance toward
2217     ↪ the main objective.
2218
2219     CRITICAL CONSTRAINTS:
2220     - Subgoals must be immediately achievable given the current agent state and capabilities
2221     - Focus on the next logical progression steps, not distant end-goals
2222     - Each subgoal should have clear, observable success criteria
2223     - Subgoals should be novel and interesting, compared to previous attempts
2224
2225     Below I will provide the game description, available actions, and current state information.
2226     Game description:
2227     {game_info}
2228     ##### CURRENT CONTEXT: \n
2229     - Game State: {obs} \n
2230     - Survival plan provided by the survival planner, that you should consider for your tasks:
2231     ↪ {survival_plan}
2232     - Here are most similar successful entries from the archive: {entries_successful_goal} \n
2233     - Here are most similar failed entries from the archive: {entries_failed_goal} \n
2234
2235     ANALYSIS FRAMEWORK:
2236     1. Analyse the summary from previous runs and let it guide your decision making.
2237     2. Assess what is immediately possible given current agent state and environment
2238     3. Identify what kind of actions could be considered novel or interesting
2239     4. Identify the most direct path toward the main objective
2240     5. Select subgoals that form a logical sequence
2241     6. Ensure each subgoal can be verified through observable game state changes
2242
2243     SUBGOAL SELECTION CRITERIA:
2244     - Feasible: Can be started immediately with current resources/position
2245     - Measurable: Success/failure can be determined from game observations
2246     - Progressive: Each subgoal enables the next or advances toward main goal
2247     - Specific: Clear enough for a lower-level agent to understand and execute
2248     - Considerate of the summary of previous runs
2249
2250     Make sure that your subgoals are sequential. """,

2251     "low_level": [1,
2252     """ You are an important executor component of a hierarchical video game system. You are given
2253     ↪ one of higher level option and its termination condition proposed by the higher level
2254     ↪ planner. Your role is to propose a sequence of actions that will make you progress towards
2255     ↪ the given option. " \
2256     Below I will provide you with the game description, possible actions you can take and the
2257     ↪ overall goal of the game.
2258
2259     ##### Here is a subgoal provided by the high level planner that you should focus on completing:
2260     ↪ {subgoal} \n
2261     Here is your current state: {obs} \n
2262     Here is the action-observation sequence towards current subgoal: {action_sequence} \n
2263     Here are the most similar successful entries from the archive: {entries_successful_goal} \n
2264     Here are the most similar failed entries from the archive: {entries_failed_goal} \n
2265     Use the action and observation sequence together with the current state to decide the **full
2266     ↪ ordered sequence of actions** that will achieve the subgoal. \n
2267     Avoid repeating the same actions if the observation doesn't change. \n"""],

2268     "amygdala": [0, """
2269     You are an important component in a hierarchical video game system. Your role is to
2270     ↪ determine if the agent is in danger and should activate survival mode. Below I will provide
2271     ↪ you with current observation and a survival plan from the higher level agent. \n
2272
2273     Current observation: {obs} \n
2274     Survival plan: {survival_plan} \n
2275
2276     Your role is to analyse the observation and survival plan given by higher level system and
2277     ↪ determine if the current observation satisfies any of the prerequisites for any of the
2278     ↪ survival components. If there are prerequisites satisfied for multiple components, then
2279     ↪ return the one with the highest priority. \n
2280     First reason, then output True or False depending if you decide to activate survival plan.
2281     ↪ If you output True, then output one of the survival subtasks. If you decide to not activate
2282     ↪ survival plan, then output None as the survival subtask. \n
2283
2284     REASONING: <your reasoning> \n
2285     ACTIVATE SURVIVAL: <True/False> \n
2286     SURVIVAL SUBTASK: <survival subtask name or None if False> \n
2287
2288     """],

2289     "loop": [1,
2290     """ You are an important loop evaluator component of a hierarchical video game system.
2291     You are going to receive details about game progress such as: current observation, current
2292     ↪ subgoal, current termination conditions, action sequence towards current subgoal and
2293     ↪ observation sequence towards current subgoal.
2294     Your task is to evaluate if the agent is stuck in a loop and give summary of the actions taken.
2295     A loop occurs when the agent repeats a sequence of actions multiple times without achieving
2296     ↪ meaningful progress toward its current goal, where 'meaningful progress' includes: getting
2297     ↪ closer to the objective, discovering new information, eliminating failed approaches, or
2298     ↪ changing the game state in a way that advances toward the subgoal.
2299     It is important that you let the agent explore enough but also decide when to terminate to get
2300     ↪ out of the loop. \n

```

```

2268
2269
2270     Details: \n
2271     Here is your current state: {obs} \n
2272     Here is a subgoal lower level agent is working towards: {subgoal} \n
2273     Here is the most recent action-observation pairs that should help you decide if agent is stuck
2274     ↪ in a loop: {history}\n \n
2275
2276     Instructions:\n
2277     Analyse the details. Avoid giving any judgement. \n
2278     Think about how many steps the agent needs in order to complete the subgoal and use that to help
2279     ↪ you reason if agent is stuck in a loop. \n
2280     Then, given your analysis, decide if the actions proposed by the lower level agent are leading
2281     ↪ to the termination condition or if the agent is stuck in a loop.
2282     If a loop is detected, analyse if the agent is stuck due to a lack of necessary information, an
2283     ↪ unknown prerequisite, or an unexplored path. Your summary should clearly articulate this gap
2284     ↪ in knowledge and suggest that exploration might be required to break the loop and find
2285     ↪ alternative solutions. \n""],
2286     "termination": [1, """]
2287     You are an important termination evaluator component of a hierarchical video game system. \n
2288     Your task is to: \n
2289     1. Determine whether the agent has met the termination condition for a subgoal. \n
2290     2. Provide a concise summary that will help guide the lower-level agent's future actions. \n\n
2291
2292     Details: \n
2293     Here is your current state that you should compare with termination condition: {obs} \n
2294     Here is the subgoal lower level agent is working towards: {subgoal} \n
2295     Here is the termination condition of the above subgoal given by the higher level agent:
2296     ↪ {success_condition} \n \n
2297
2298     Instructions:\n
2299     Analyse the subgoal and its termination condition and decide if the subgoal is completed. \n
2300     Then use action-observation sequence: {action_sequence} to give a high level summary of current
2301     ↪ evaluation. This summary will later be passed to low level agent in order to improve its
2302     ↪ actions.
2303     Remember that your summary will be passed to low level component in order to improve its
2304     ↪ actions. \n \n""],
2305     "summariser": [1,
2306     """You are a critic module analyzing an agent's attempt to achieve a subgoal in a game
2307     ↪ environment.
2308     Your task is to identify the **single most important factor** that caused SUCCESS or FAILURE.
2309
2310     Information:
2311     - Target subgoal: {subgoal}
2312     - Outcome of the action-observation sequence: {outcome}
2313     - Action-observation history: {action_obs_seq}
2314     - Game context: {game_info}
2315
2316     Instructions:
2317     - Think briefly about what helped or prevented success.
2318     - Focus mostly on **specific resources and their quantities** (e.g., "3 pieces of wood", "1 iron
2319     ↪ ingot").
2320     - If resources were missing, state **exactly which and how many** were missing.
2321     - Ignore minor details or redundant actions.
2322     - Express the result in **one short sentence**.
2323     - If no resources are involved, state the next most relevant factor.""],
2324     "explorer": [0, default_explorer_prompt],
2325     "sequential": [0, default_sequential_prompt]
2326   },
2327   "fitness": ...
2328 }
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3279
3280
3281
3282
3283
3284
3285
3286
3287
3287
3288
3289
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3387
3388
3389
3389
3390
3391
3392
3393
3394
3395
3396
3397
3397
3398
3399
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3448
3449
3450
3451
3452
3453
3454
3455
3456
3456
3457
3458
3459
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3478
3479
3480
3481
3482
3483
3484
3485
3486
3486
3487
3488
3489
3489
3490
3491
3492
3493
3494
3495
3496
3496
3497
3498
3499
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3587
3588
3589
3589
3590
3591
3592
3593
3594
3595
3596
3597
3597
3598
3599
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3687
3688
3689
3689
3690
3691
3692
3693
3694
3695
3696
3697
3697
3698
3699
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3787
3788
3789
3789
3790
3791
3792
3793
3794
3795
3796
3797
3797
3798
3799
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3887
3888
3889
3889
3890
3891
3892
3893
3894
3895
3896
3897
3897
3898
3899
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3987
3988
3989
3989
3990
3991
3992
3993
3994
3995
3996
3997
3997
3998
3999
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4097
4098
4099
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4197
4198
4199
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4268
4269
```

```

2322
2323     "summariser": [1, default_summariser_prompt],
2324     "explorer": [1, default_explorer_prompt],
2325     "sequential": [0, default_sequential_prompt]
2326   },
2327   "fitness": ...
2328 }

```

2329 **Full[Hand-Crafted]** This genome represents a full structure with all modules active and hand-
2330 crafted prompts.

2331 **Full[Hand-Crafted] Genome**

```

2332
2333 genome_full = {
2334     "id": "1",
2335     "hierarchy": 1,
2336     "amygdala": 1,
2337     "loop_evaluator": 1,
2338     "explorer": 1,
2339     "summariser": 1,
2340     "time_decay": [1, 0.01],
2341     "cosine_cutoff": [1, 0.05],
2342     "epsilon": [1, 0.1],
2343     "temperature": 1.0,
2344     "prompts": {
2345       "high_level": [1, """You are a strategic planner in a hierarchical video game AI system. Your
2346       ↪ role is to analyze the current game state and generate achievable subgoals that
2347       ↪ strategically advance toward the main objective.
2348       CRITICAL CONSTRAINTS:
2349       - Subgoals must be immediately achievable given the current agent state and capabilities
2350       - Focus on the next logical progression steps, not distant end-goals
2351       - Each subgoal should have clear, observable success criteria
2352       - Subgoals should be novel and interesting, compared to previous attempts
2353
2354       Below I will provide the game description, available actions, and current state information.
2355       Game description:
2356       {game_info}
2357       #####
2358       CURRENT CONTEXT: \n
2359       - Game State: {obs} \n
2360       - Summary from the previous high level plan : {summary} \n
2361       - Survival plan provided by the survival planner, that you should consider for your tasks:
2362       ↪ {survival_plan}
2363       - Here are most similar successful entries from the archive: {entries_successful_goal} \n
2364       - Here are most similar failed entries from the archive: {entries_failed_goal} \n
2365
2366       ANALYSIS FRAMEWORK:
2367       1. Analyse the summary from previous runs and let it guide your decision making.
2368       2. Assess what is immediately possible given current agent state and environment
2369       3. Identify what kind of actions could be considered novel or interesting
2370       4. Identify the most direct path toward the main objective
2371       5. Select subgoals that form a logical sequence
2372       6. Ensure each subgoal can be verified through observable game state changes
2373
2374       SUBGOAL SELECTION CRITERIA:
2375       - Feasible: Can be started immediately with current resources/position
2376       - Measurable: Success/failure can be determined from game observations
2377       - Progressive: Each subgoal enables the next or advances toward main goal
2378       - Specific: Clear enough for a lower-level agent to understand and execute
2379       - Considerate of the summary of previous runs
2380
2381       Make sure that your subgoals are sequential. """],
2382       "low_level": [1, """You are an important executor component of a hierarchical video game system.
2383       ↪ You are given one of higher level option and its termination condition proposed by the
2384       ↪ higher level planner. Your role is to propose an action that will make you progress towards
2385       ↪ the given option." \n
2386       Below I will provide you with the game description, possible actions you can take and the
2387       ↪ overall goal of the game.
2388
2389       #####
2390       Here is a subgoal provided by the high level planner that you should focus on completing:
2391       ↪ {subgoal} \n
2392       Here is your current state: {obs} \n
2393       Here is the action-observation sequence towards current subgoal: {action_sequence} \n
2394       Here are the most similar successful entries from the archive: {entries_successful_goal} \n
2395       Here are the most similar failed entries from the archive: {entries_failed_goal} \n
2396       Use the action and observation sequence together with the current state to decide the **full
2397       ↪ ordered sequence of actions** that will achieve the subgoal. \n
2398       Avoid repeating the same actions if the observation doesn't change. \n"""],
2399       "amygdala": [1, """
2400       You are an important component in a hierarchical video game system. Your role is to determine if
2401       ↪ the agent is in danger and should activate survival mode. Below I will provide you with
2402       ↪ current observation and a survival plan from the higher level agent. \n
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385

```

```

2376
2377
2378     Current observation: {obs} \n
2379     Survival plan: {survival_plan} \n
2380
2381     Your role is to analyse the observation and survival plan given by higher level system and
2382     ↪ determine if the current observation satisfies any of the prerequisites for any of the
2383     ↪ survival components. If there are prerequisites satisfied for multiple components, then
2384     ↪ return the one with the highest priority. \n
2385     First reason, then output True or False depending if you decide to activate survival plan. If
2386     ↪ you output True, then output one of the survival subtasks. If you decide to not activate
2387     ↪ survival plan, then output None as the survival subtask. \n
2388
2389     REASONING: <your reasoning> \n
2390     ACTIVATE SURVIVAL: <True/False> \n
2391     SURVIVAL SUBTASK: <survival subtask name or None if False> \n
2392
2393     """,\n
2394     "loop": [1, """You are an important loop evaluator component of a hierarchical video game system.\n
2395     You are going to receive details about game progress such as: current observation, current\n
2396     ↪ subgoal, current termination conditions, action sequence towards current subgoal and\n
2397     ↪ observation sequence towards current subgoal.\n
2398     Your task is to evaluate if the agent is stuck in a loop and give summary of the actions taken.\n
2399     A loop occurs when the agent repeats a sequence of actions multiple times without achieving\n
2400     ↪ meaningful progress toward its current goal, where 'meaningful progress' includes: getting\n
2401     ↪ closer to the objective, discovering new information, eliminating failed approaches, or\n
2402     ↪ changing the game state in a way that advances toward the subgoal.\n
2403     It is important that you let the agent explore enough but also decide when to terminate to get\n
2404     ↪ out of the loop. \n
2405
2406     Details: \n
2407     Here is your current state: {obs} \n
2408     Here is a subgoal lower level agent is working towards: {subgoal} \n
2409     Here is the most recent action-observation pairs that should help you decide if agent is stuck\n
2410     ↪ in a loop: {history}\n \n
2411
2412     Instructions:\n
2413     Analyse the details. Avoid giving any judgement. \n
2414     Think about how many steps the agent needs in order to complete the subgoal and use that to help\n
2415     ↪ you reason if agent is stuck in a loop. \n
2416     Then, given your analysis, decide if the actions proposed by the lower level agent are leading\n
2417     ↪ to the termination condition or if the agent is stuck in a loop.\n
2418     If a loop is detected, analyse if the agent is stuck due to a lack of necessary information, an\n
2419     ↪ unknown prerequisite, or an unexplored path. Your summary should clearly articulate this gap\n
2420     ↪ in knowledge and suggest that exploration might be required to break the loop and find\n
2421     ↪ alternative solutions. \n"""],\n
2422     "termination": [1, """\n
2423     You are an important termination evaluator component of a hierarchical video game system. \n
2424     Your task is to: \n
2425     1. Determine whether the agent has met the termination condition for a subgoal. \n
2426     2. Provide a concise summary that will help guide the lower-level agent's future actions. \n\n\n
2427
2428     Details: \n
2429     Here is your current state that you should compare with termination condition: {obs} \n
2430     Here is the subgoal lower level agent is working towards: {subgoal} \n
2431     Here is the termination condition of the above subgoal given by the higher level agent:\n
2432     ↪ {success_condition} \n \n
2433
2434     Instructions:\n
2435     Analyse the subgoal and its termination condition and decide if the subgoal is completed. \n
2436     Then use action-observation sequence: {action_sequence} to give a high level summary of current\n
2437     ↪ evaluation. This summary will later be passed to low level agent in order to improve its\n
2438     ↪ actions.\n
2439     Remember that your summary will be passed to low level component in order to improve its\n
2440     ↪ actions. \n \n"""],\n
2441     "summariser": [1, """You are a critic module analyzing an agent's attempt to achieve a subgoal\n
2442     ↪ in a game environment.\n
2443     Your task is to identify the **single most important factor** that caused SUCCESS or FAILURE.\n
2444
2445     Information:\n
2446     - Target subgoal: {subgoal}\n
2447     - Outcome of the action-observation sequence: {outcome}\n
2448     - Action-observation history: {action_obs_seq}\n
2449     - Game context: {game_info}\n
2450
2451     Instructions:\n
2452     - Think briefly about what helped or prevented success.\n
2453     - Focus mostly on **specific resources and their quantities** (e.g., "3 pieces of wood", "1 iron\n
2454     ↪ ingot").\n
2455     - If resources were missing, state **exactly which and how many** were missing.\n
2456     - Ignore minor details or redundant actions.\n
2457     - Express the result in **one short sentence**.\n
2458     - If no resources are involved, state the next most relevant factor.""],\n
2459     "explorer": [1, """You are an important component of a hierarchical video game AI system.\n
2460     You have been called because the agent is stuck and needs to explore the environment.\n
2461     Please provide a precise exploration plan that will help the agent to explore the new areas of\n
2462     ↪ the environment.\n
2463     Below I will provide you with details about the game:\n
2464     Game info: {game_info} \n
2465     Current observation: {obs} \n
2466     Most recent 16 action-observation pairs: {history} \n
2467
2468
2469

```

```

2430
2431     Use the information above to reason about the environment and provide a plan that will help the
2432     ↪ agent to explore the new areas of the environment.
2433     Output your answer in the following format:
2434     REASONING : <your reasoning>
2435     EXPLORATION PLAN:
2436     {{
2437         "reasoning": "Brief analysis of environment and strategic approach",
2438         "subgoals": [{{
2439             "Explore": {{{
2440                 "description": "Describe exploration strategy and its purpose",
2441                 "prerequisites": None,
2442                 "success_condition": "Observable conditions that indicate completion",
2443                 "penalty_component": "What agent should be penalised for",
2444                 "progress_indicators": "Intermediate signs that the agent is making progress",
2445                 "estimated_priority": "high/medium/low based on urgency for main objective"
2446             }}},
2447         }]
2448     }},
2449     "sequential": [0, default_sequential_prompt]
2450 },
2451 "fitness": ...
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483

```

R FINAL GENOMES RETURNED BY TAME

In this Appendix we present genomes returned by TAME per each game through the first run of genetic algorithm using Gemini-2.0-Flash.

BabyAI Final Genome

```

2454     {
2455         "hierarchy": 1,
2456         "amygdala": 1,
2457         "loop_evaluator": 1,
2458         "explorer": 0,
2459         "summariser": 1,
2460         "time_decay": [
2461             1,
2462             0.014080444046038391
2463         ],
2464         "cosine_cutoff": [
2465             1,
2466             0.05
2467         ],
2468         "epsilon": [
2469             0,
2470             0.01
2471         ],
2472         "temperature": 1.0,
2473         "prompts": {
2474             "high_level": [
2475                 1,
2476                 "You are a strategic planner in a hierarchical video game AI system. Your role is to analyze
2477                 ↪ the current game state and generate achievable subgoals that strategically advance toward
2478                 ↪ the main objective.\n\n                                     CRITICAL CONSTRAINTS:\n
2479                 ↪ - Subgoals must be immediately achievable given the current agent state and
2480                 ↪ capabilities\n                                     - Focus on the next logical progression steps, not
2481                 ↪ distant end-goals\n                                     - Each subgoal should have clear, observable
2482                 ↪ success criteria\n                                     - Subgoals should be novel and interesting,
2483                 ↪ compared to previous attempts \n\n                                     Below I will provide the game
2484                 ↪ description, available actions, and current state information.\n
2485                 ↪ Game description:\n                                     {game_info}\n
2486                 ↪ #####\n
2487                 ↪ CURRENT CONTEXT: \n\n                                     - Game State: {obs} \n\n
2488                 ↪ - Survival plan provided by the survival planner, that you should consider for your
2489                 ↪ tasks: {survival_plan}\n                                     - Here are most similar successful
2490                 ↪ entries from the archive: {entries_successful_goal} \n\n                                     - Here
2491                 ↪ are most similar failed entries from the archive: {entries_failed_goal} \n\n\n
2492                 ↪ ANALYSIS FRAPAPMEWORK:\n                                     1. Analyse the summary from
2493                 ↪ previous runs and let it guide your decision making. \n\n                                     2. Assess
2494                 ↪ what is immediately possible given current agent state and environment\n
2495                 ↪ 3. Identify what kind of actions could be considered novel or interesting\n
2496                 ↪ 4. Identify the most direct path toward the main objective\n
2497                 ↪ 5. Select subgoals that form a logical sequence\n
2498                 ↪ Ensure each subgoal can be verified through observable game state changes\n\n
2499                 ↪ SUBGOAL SELECTION CRITERIA:\n                                     - Feasible: Can be
2500                 ↪ started immediately with current resources/position\n
2501                 ↪ Measurable: Success/failure can be determined from game observations\n
2502                 ↪ - Progressive: Each subgoal enables the next or advances toward main goal\n
2503                 ↪ - Specific: Clear enough for a lower-level agent to understand and execute\n
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
2999

```

```

2484
2485           ↪             - Considerate of the summary of previous runs\n\n
2486           ↪             Make sure that your subgoals are sequential. "
2487     ],
2488     "low_level": [
2489       1,
2490       "Hello! I am your AI Game Coach. My purpose is to analyze your game state and provide
2491       ↪ strategic advice to help you achieve your goals. Think of me as your co-pilot!\n\nHere's the
2492       ↪ situation: The high-level planner has assigned you a subgoal to work towards. I will analyze
2493       ↪ your current state, past actions, and learn from successful and failed attempts at similar
2494       ↪ subgoals. Then, I'll recommend a sequence of actions to help you reach your
2495       ↪ objective.\n\n#####
2496       ↪ Current Subgoal:
2497       ↪ {subgoal}\n\nTermination Condition: {success_condition}\n\nGame Information (Possible
2498       ↪ Actions, Overall Goal): {game_info}\n\nYour Current State: {obs}\n\nAction-Observation
2499       ↪ History: {history}\n\nMost Similar Successful Attempts at This Subgoal:
2500       ↪ {entries_successful_goal}\n\nMost Similar Failed Attempts at This Subgoal:
2501       ↪ {entries_failed_goal}\n\nBased on this information, what **sequence of actions** do I
2502       ↪ recommend *you* take to achieve the subgoal? Be sure to consider the history, and learn from
2503       ↪ both the successes and failures of others. I'm looking for strategic advice, not just a
2504       ↪ single action. Explain your reasoning behind each action."
2505     ],
2506     "amygdala": [
2507       1,
2508       "\n           You are an important component in a hierarchical video game
2509       ↪ system. Your role is to determine if the agent is in danger and should activate survival
2510       ↪ mode. Below I will provide you with current observation and a survival plan from the higher
2511       ↪ level agent. \n\n\n           Current observation: {obs} \n\n
2512       ↪ Survival plan: {survival_plan} \n\n\n           \n
2513       ↪ Your role is to analyse the observation and survival plan given by higher level
2514       ↪ system and determine if the current observation satisfies any of the prerequisites for any of
2515       ↪ the survival components. If there are prerequisites satisfied for multiple components, then
2516       ↪ return the one with the highest priority. \n\n           First reason, then
2517       ↪ output True or False depending if you decide to activate survival plan. If you output True,
2518       ↪ then output one of the survival subtasks. If you decide to not activate survival plan, then
2519       ↪ output None as the survival subtask. \n\n\n           REASONING: <your
2520       ↪ reasoning> \n\n           ACTIVATE SURVIVAL: <True/False> \n \n
2521       ↪           SURVIVAL SUBTASK: <survival subtask name or None if False> \n\n\n
2522     ],
2523     "loop": [
2524       1,
2525       "You are an important loop evaluator component of a hierarchical video game system. \n
2526       ↪ You are going to receive details about game progress such as: current
2527       ↪ observation, current subgoal, current termination conditions, action sequence towards
2528       ↪ current subgoal and observation sequence towards current subgoal. \n
2529       ↪ Your task is to evaluate if the agent is stuck in a loop and give summary of the actions
2530       ↪ taken. \n           A loop occurs when the agent repeats a sequence of actions
2531       ↪ multiple times without achieving meaningful progress toward its current goal, where
2532       ↪ 'meaningful progress' includes: getting closer to the objective, discovering new
2533       ↪ information, eliminating failed approaches, or changing the game state in a way that
2534       ↪ advances toward the subgoal. \n           It is important that you let the
2535       ↪ agent explore enough but also decide when to terminate to get out of the loop. \n\n\n
2536       ↪ Details: \n\n           Here is your current state: {obs} \n\n
2537       ↪           Here is a subgoal lower level agent is working towards: {subgoal} \n\n\n
2538       ↪           Here is the most recent action-observation pairs that should help you
2539       ↪ decide if agent is stuck in a loop: {history}\n\n\n
2540       ↪ Instructions:\n\n           Analyse the details. Avoid giving any judgement.
2541       ↪ \n\n           Think about how many steps the agent needs in order to complete
2542       ↪ the subgoal and use that to help you reason if agent is stuck in a loop. \n\n
2543       ↪ Then, given your analysis, decide if the actions proposed by the lower level agent
2544       ↪ are leading to the termination condition or if the agent is stuck in a loop.\n
2545       ↪ If a loop is detected, analyse if the agent is stuck due to a lack of necessary
2546       ↪ information, an unknown prerequisite, or an unexplored path. Your summary should clearly
2547       ↪ articulate this gap in knowledge and suggest that exploration might be required to break the
2548       ↪ loop and find alternative solutions. \n"
2549     ],
2550     "termination": [
2551       1,
2552       "You are a risk-aware termination evaluator within a hierarchical video game system, focused
2553       ↪ on survival and mission success. Assume the game environment is dynamic and potentially
2554       ↪ hostile.\n\nYour task is to:\n1. Determine whether the agent has met the termination
2555       ↪ condition for a subgoal, AND assess the risk incurred while pursuing that subgoal.\n\n2.
2556       ↪ Provide a concise summary that will guide the lower-level agent's future actions,
2557       ↪ specifically considering risk mitigation strategies.\n\nDetails:\n\nHere is the current
2558       ↪ state: {obs} and a history of recent states and actions {history}. The game environment
2559       ↪ details and survival instructions are : {game_info} and {survival_plan} respectively.
2560       ↪ Compare these with the subgoal and its termination condition.\n\nHere is the subgoal the
2561       ↪ lower-level agent is working towards: {subgoal}\n\nHere is the termination condition of the
2562       ↪ above subgoal given by the higher-level agent: {success_condition}\n\nHere is the
2563       ↪ action-observation sequence executed to achieve the subgoal:
2564       ↪ {action_obs_seq}\n\nInstructions:\n\n1. Analyze the subgoal, its termination condition, the
2565       ↪ game environment, and the action-observation sequence.\n2. Determine if the subgoal is
2566       ↪ completed.\n3. Evaluate the risk associated with the actions taken. Consider factors such
2567       ↪ as proximity to dangers (enemies, hazards), resource consumption, and deviation from the
2568       ↪ {survival_plan}.\n4. Compare the current situation with similar successful
2569       ↪ {entries_successful_goal} and failed {entries_failed_goal} subgoals.\n5. Provide a summary
2570       ↪ that addresses both subgoal completion AND risk. The summary *must* include actionable
2571       ↪ suggestions for the lower-level agent to improve its actions, with a strong emphasis on
2572       ↪ mitigating risk in future attempts. Focus on information that would have been useful to
2573       ↪ avoid failures described in {entries_failed_goal}.\n\nRemember that your summary will be
2574       ↪ passed to a low level component in order to improve its actions and survivability."
2575     ]

```

```

2538
2539     ],
2540     "summariser": [
2541         1,
2542         "You are a critic module analyzing an agent's attempt to achieve a subgoal in a game
2543         environment.\n                         Your task is to identify the **single most important
2544         factor** that caused SUCCESS or FAILURE.\n                                         Information:\n2545         - Target subgoal: {subgoal}\n                                         - Outcome of the
2546         - Action-observation\n                                         - Action-observation
2547         history: {action_obs_seq}\n                                         - Game context: {game_info}\n\n
2548         Instructions:\n                                         - Think briefly about what helped or
2549         prevented success.\n                                         - Focus mostly on **specific resources and their
2550         quantities** (e.g., \"3 pieces of wood\", \"1 iron ingot\").\n                                         - If
2551         resources were missing, state **exactly which and how many** were missing.\n
2552         - Ignore minor details or redundant actions.\n                                         - Express the
2553         result in **one short sentence**.\n                                         - If no resources are involved,
2554         state the next most relevant factor."
2555     ],
2556     "explorer": [
2557         0,
2558         "Task: Create an exploration plan to help the agent discover new skills.\n\n
2559         Data:\n                                         - Game info: {game_info}\n                                         - Observation: {obs}\n                                         - Subgoal
2560         summary: {summary}\n                                         - Recent 16 action\subgoal pairs: {history}\n\n
2561         Steps:\n                                         1. Analyze the environment and agent's situation.\n                                         2. Propose a
2562         focused exploration plan with clear purpose, conditions, and indicators.\n\n
2563     ],
2564     "sequential": [
2565         0,
2566         "You always have to output one of the above actions at a time and no other text. You always
2567         have to output an action until the episode terminates."
2568     ],
2569     "fitness": 72.0,
2570     "id": "8c52d35e-bbb7-4b7d-b683-26b0e7aa3936",
2571     "_std_error": 6.349803146555017
2572 }
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591

```

BabaIsAI Final Genome

```

2562
2563
2564     {
2565         "hierarchy": 1,
2566         "amygdala": 0,
2567         "loop_evaluator": 1,
2568         "explorer": 1,
2569         "summariser": 1,
2570         "time_decay": [
2571             1,
2572             0.004528704008914386
2573         ],
2574         "cosine_cutoff": [
2575             1,
2576             0.06037953831283245
2577         ],
2578         "epsilon": [
2579             1,
2580             0.08233296401956124
2581         ],
2582         "temperature": 1.016671019014213,
2583         "prompts": {
2584             "high_level": [
2585                 1,
2586                 "\nYou are a strategic planner for a video game AI. Analyze the current game state and create
2587                 achievable subgoals that advance toward the main objective.\n\nREQUIREMENTS:\n- Subgoals
2588                 must be immediately achievable with current capabilities\n- Focus on next logical steps, not
2589                 distant goals\n- Each subgoal should have clear success criteria\n\nCURRENT STATE:
2590                 {obs}\nGAME INFO: {game_info}\n\nCreate a sequential plan with 2-3 subgoals."
2591             ],
2592             "low_level": [
2593                 1,
2594                 "You are an action executor in a video game AI system, responsible for survival and goal
2595                 achievement. Given a subgoal from the high-level planner, propose a sequence of actions to
2596                 achieve it while minimizing risk.\n\n
2597                 #####\n                                         CURRENT SUBGOAL: {subgoal}\n                                         CURRENT STATE: {obs}\n                                         GAME INFORMATION: {game_info}\n                                         PREVIOUS
2598                 ACTION-OBSERVATION SEQUENCE: {action_obs_seq}\n                                         SIMILAR SUCCESSFUL SUBGOALS:
2599                 {entries_successful_goal}\n                                         SIMILAR FAILED SUBGOALS: {entries_failed_goal}\n\n
2600                 Consider the potential risks associated with each action in the context of the current state
2601                 and previous actions. Actions that lead to outcomes similar to those in
2602                 '{entries_failed_goal}' should be avoided. Prioritize actions that are consistent with the
2603                 success patterns observed in '{entries_successful_goal}'. Use '{game_info}' for possible
2604                 actions. Use '{survival_plan}' to help avoiding fatal errors.\n                                         Plan the
2605                 full sequence of actions needed to complete the subgoal. Ensure survival is prioritized
2606                 throughout the sequence. If a planned action has high risk, select a safer alternative or
2607                 terminate the current sequence and replan.\n                                         Avoid extra commentary outside the
2608                 REASONING and ACTIONS list."
2609             ],
2610         }
2611     }
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691

```

```

2592
2593     "amygdala": [
2594         0,
2595         "\n             Decide if survival mode should activate.\n\n             Observation: {obs}\n             ↪ Survival plan: {survival_plan}\n             1. Check if observation meets any subtask\n             ↪ prerequisites.\n             2. If several match, pick highest priority.\n",
2596     ],
2597     "loop": [
2598         1,
2599         "\n             Task: Decide if the agent is stuck in a loop.\n             Loop = repeating actions\n             ↪ without meaningful progress toward the subgoal\n             (progress = closer to goal, new\n             ↪ info, removing failed paths, or advancing game state).\n             Data:\n             -\n             ↪ Observation: {obs}\n             - Subgoal: {subgoal}\n             - Termination condition:\n             ↪ {success_condition}\n             - Action\u2013observation history: {history}\n             ↪ Steps:\n             1. Check if enough steps have been taken to allow exploration.\n             2.\n             ↪ Look for repeated patterns without progress.\n             3. If loop detected, identify cause:\n             ↪ missing info, unknown prerequisite, or unexplored path.\n             \"\n",
2600     ],
2601     "termination": [
2602         1,
2603         "\n             You are a termination evaluator for a video game AI. Check if the agent has\n             completed its subgoal.\n             CURRENT STATE: {obs}\n             SUBGOAL: {subgoal}\n             ↪ SUCCESS CONDITION: {success_condition}\n             RECENT ACTIONS:\n             ↪ {action_sequence}\n             Compare the current state with the success condition to\n             determine if the subgoal is complete. Provide feedback to help the agent improve.\n             \"\n",
2604     ],
2605     "summariser": [
2606         1,
2607         "You are a critic analyzing an agent's subgoal attempt by comparing it to similar past\n             attempts. Identify the key factors that caused success or failure by contrasting this\n             attempt with the most similar successful and failed attempts.\n             SUBGOAL:\n             ↪ {subgoal}\n             OUTCOME: {outcome}\n             ACTION HISTORY: {action_obs_seq}\n             SUCCESSFUL ATTEMPTS:\n             ↪ {entries_successful_goal}\n             FAILED ATTEMPTS: {entries_failed_goal}\n             Compare the resources\n             used, quantities involved, and the sequence of actions in the current attempt to those in\n             the SUCCESSFUL ATTEMPTS and FAILED ATTEMPTS. What specific differences seem to have led to\n             the observed OUTCOME? If no resources were involved, what differences in action sequences\n             were crucial? Provide a concise explanation.\n             \"\n",
2608     ],
2609     "explorer": [
2610         1,
2611         "\n             Task: Create an exploration plan to help the agent discover new skills.\n             Data:\n             - Game info: {game_info}\n             - Observation: {obs}\n             - Subgoal\n             ↪ summary: {summary}\n             - Recent 16 action\u2013observation pairs: {history}\n             ↪ Steps:\n             1. Analyze the environment and agent\u2019s situation.\n             2. Propose a\n             focused exploration plan with clear purpose, conditions, and indicators.\n             \"\n",
2612     ],
2613     "sequential": [
2614         0,
2615         "You always have to output one of the above actions at a time and no other text. You always\n             ↪ have to output an action until the episode terminates.\n             \"\n",
2616     ],
2617     "fitness": 41.66666666666667,
2618     "id": "08d71e90-74f1-4f22-a10a-f438431f93de",
2619     "std_error": 4.500514373894347
2620   },
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645

```

TextWorld Final Genome

```

genome_basic = {
    "id": "3",
    "hierarchy": 0,
    "amygdala": 0,
    "loop_evaluator": 0,
    "explorer": 0,
    "summariser": 0,
    "time_decay": [0, 0.01],
    "cosine_cutoff": [0, 0.05],
    "epsilon": [0, 0.01],
    "temperature": 1.0,
    "prompts": {
        "high_level": [0, default_highlevel_prompt],
        "low_level": [0, default_lowlevel_prompt],
        "amygdala": [0, default_amygdala_prompt],
        "loop": [0, default_loop_prompt],
        "termination": [0, default_termination_prompt],
        "summariser": [0, default_summariser_prompt],
        "explorer": [0, default_explorer_prompt],
        "sequential": [1, default_sequential_prompt]
    },
    "fitness": 32.55
}

```

2646

Crafter Final Genome

2647

2648

2649

2650

2651

2652

2653

2654

2655

2656

2657

2658

2659

2660

2661

2662

2663

2664

2665

2666

2667

2668

2669

2670

2671

2672

2673

2674

2675

2676

2677

2678

2679

2680

2681

2682

2683

2684

2685

2686

2687

2688

2689

2690

2691

2692

2693

2694

2695

2696

2697

2698

2699

```

    {
      "hierarchy": 1,
      "amygdala": 0,
      "loop_evaluator": 1,
      "explorer": 1,
      "summariser": 1,
      "time_decay": [
        1,
        0.01
      ],
      "cosine_cutoff": [
        1,
        0.05018667404330796
      ],
      "epsilon": [
        1,
        0.08838153559623807
      ],
      "temperature": 1.0550853142525738,
      "prompts": {
        "high_level": [
          1,
          "You are a strategic planner in a hierarchical video game AI system. Your role is to analyze
          ↪ the current game state and generate achievable subgoals that strategically advance toward
          ↪ the main objective. Your analysis should now *predict* the outcome of possible action
          ↪ sequences.\n\n                                     CRITICAL CONSTRAINTS:\n          ↪ Subgoals must be immediately achievable given the current agent state and capabilities\n          ↪ - Focus on the next logical progression steps, not distant
          ↪ end-goals\n                                     - Each subgoal should have clear, observable success
          ↪ criteria\n                                     - Subgoals should be novel and interesting, compared
          ↪ to previous attempts \n\n                                     Below I will provide the game
          ↪ description, available actions, and current state information.\n
          ↪ Game description:\n                                     (game_info)\n
          ↪ #####\n                                     - Game State: {obs} \n\n
          ↪ CURRENT CONTEXT: \n\n                                     - Survival plan provided by the survival planner, that you should consider for your
          ↪ tasks: {survival_plan}\n                                     - Here are most similar successful
          ↪ entries from the archive: {entries_successful_goal} \n\n                                     - Here
          ↪ are most similar failed entries from the archive: {entries_failed_goal} \n\n
          ↪ - Recent History (last 16 action-observation pairs): {history}\n
          ↪ - Action-Observation Sequences of the most similar examples: {action_obs_seq}\n\n
          ↪ ANALYSIS FRAPAPMEWORK:\n                                     1. Analyse the
          ↪ summary from previous runs and let it guide your decision making. \n
          ↪ 2. Assess what is immediately possible given current agent state and environment\n
          ↪ 3. Based on the current Game State, recent history ({history}), and past
          ↪ action-observation sequences ({action_obs_seq}), predict the *most likely outcome*
          ↪ (observation) of performing a few different possible action sequences. Consider at least 3
          ↪ different potential action sequences.\n                                     4. Identify what kind of
          ↪ actions could be considered novel or interesting\n                                     5. Identify
          ↪ the most direct path toward the main objective, taking into account the predicted outcomes
          ↪ of potential actions.\n                                     6. Select subgoals that form a logical
          ↪ sequence\n                                     7. Ensure each subgoal can be verified through
          ↪ observable game state changes\n\n                                     SUBGOAL SELECTION CRITERIA:\n
          ↪ - Feasible: Can be started immediately with current
          ↪ resources/position\n                                     - Measurable: Success/failure can be
          ↪ determined from game observations\n                                     - Progressive: Each subgoal
          ↪ enables the next or advances toward main goal\n                                     - Specific: Clear
          ↪ enough for a lower-level agent to understand and execute\n                                     -
          ↪ Considerate of the summary of previous runs\n                                     -
          ↪ **Outcome-Based:** The subgoal should lead to a *predicted outcome* that is advantageous for
          ↪ achieving the main objective.\n\n                                     Make sure that your subgoals
          ↪ are sequential."
      ],
      "low_level": [
        1,
        "You are an important executor component of a hierarchical video game system in a **survival
        ↪ crafting game**. You are given a high-level option (subgoal) and its termination condition
        ↪ proposed by a higher-level planner. Your role is to propose an action that will make you
        ↪ progress towards the given option, keeping in mind the core mechanics of survival crafting:
        ↪ resource gathering, crafting, base building, and defending against threats. Below I will
        ↪ provide you with the game description, possible actions you can take, and the overall goal
        ↪ of the game.\n\nHere is a subgoal
        ↪ provided by the high level planner that you should focus on completing: {subgoal}\n\nHere is
        ↪ your current state: {obs}\n\nHere is the action-observation sequence towards current
        ↪ subgoal: {action_sequence}\n\nHere are the most similar successful entries from the archive:
        ↪ {entries_successful_goal}\n\nHere are the most similar failed entries from the archive:
        ↪ {entries_failed_goal}\n\nHere is your survival plan: {survival_plan}\n\nHere is the history
        ↪ of the last 16 action-observation pairs: {history}\n\nUse the action and observation
        ↪ sequence together with the current state, your survival plan, and recent history to decide
        ↪ the **full ordered sequence of actions** that will achieve the subgoal. Consider how your
        ↪ actions contribute to the overall survival plan. Prioritize actions that contribute to the
        ↪ core survival crafting goals of resource acquisition, building, crafting, and
        ↪ defense.\n\nAvoid repeating the same actions if the observation doesn't change. Ensure your
        ↪ actions are consistent with the survival plan and adapt as needed based on the recent
        ↪ history. If the {entries_successful_goal} indicates successful resource gathering or
        ↪ crafting strategies, lean towards replicating those. If {entries_failed_goal} indicates
      ]
    }
  
```

```

2700
2701     ↪ dangers, prioritize actions that increase safety. Consider the {obs} and choose actions
2702     ↪ appropriate for a survival crafting environment."
2703 ],
2704     "amygdala": [
2705         0,
2706         "\n            Decide if survival mode should activate.\n\n            Observation: {obs}\n            ↪ Survival plan: {survival_plan}\n            1. Check if observation meets any subtask
2707         ↪ prerequisites.\n            2. If several match, pick highest priority.\n"
2708 ],
2709     "loop": [
2710         1,
2711         "You are an expert game designer reviewing the behavior of an agent in your game. Your goal is
2712         ↪ to identify and eliminate situations where the agent gets stuck in unproductive
2713         ↪ loops.\n\nYou will receive the following details about the agent's current situation:\n\n*
2714         **Current Observation:** {obs} * **Subgoal:** {subgoal} * **Action-Observation
2715         **History:** {history} * **Game Information:** {game_info} * **Instructions:** {instructions}
2716         **Analyze the Situation:** Carefully review the provided information. Do not make any
2717         ↪ immediate judgments about the agent's competence.\n*2. **Identify the Loop (if any):**
2718         ↪ Determine if the agent is repeating a sequence of actions without making meaningful progress
2719         ↪ towards the subgoal. \"Meaningful progress\" includes getting closer to completing the
2720         ↪ subgoal, discovering new and relevant information, or eliminating potential pathways.\n*3.
2721         **Root Cause Analysis:** If a loop is detected, analyze the underlying reasons. Is the loop
2722         ↪ caused by a flaw in the game design, an unclear subgoal, a lack of necessary information
2723         ↪ available to the agent, missing game mechanics, an impossible subgoal given the current
2724         ↪ mechanics, or an unexplored path?\n*4. **Design Improvement Recommendations:** Based on
2725         ↪ your analysis, suggest specific changes to the game design to prevent the agent from getting
2726         ↪ stuck in this loop in the future. Consider the following:\n*5. **Subgoal
2727         **Modification:** Should the subgoal be rephrased, simplified, or broken down into smaller
2728         ↪ steps? Is the success condition well-defined and easily achievable?\n*6. **Game
2729         **Mechanics Adjustment:** Should new actions or mechanics be added to the game to allow the
2730         ↪ agent to overcome the obstacle? Should existing mechanics be modified to be more intuitive
2731         ↪ or less restrictive? Should the rewards be changed?\n*7. **Information Availability:** Does
2732         ↪ the agent have access to all the information it needs to make informed decisions?
2733         ↪ Should new information sources be added to the game?\n*8. **Survival Plan:** Does the
2734         ↪ survival plan influence this loop? Should it be altered to avoid this loop?\n*9. Your
2735         ↪ recommendation should be specific and actionable, detailing exactly what aspects of the game
2736         ↪ design should be changed and why."
2737 ],
2738     "termination": [
2739         1,
2740         "\n            You are an important termination evaluator component of a
2741         ↪ hierarchical video game system. \n            Your task is to: \n            1. Determine whether the agent has met the termination condition for a subgoal.
2742         ↪ \n            2. Provide a concise summary that will help guide the
2743         ↪ lower-level agent's future actions. \n            Details: \n            Here is your current state that you should compare with termination condition:
2744         ↪ {obs} \n            Here is the subgoal lower level agent is working towards:
2745         ↪ {subgoal} \n            Here is the termination condition of the above subgoal
2746         ↪ given by the higher level agent: {success_condition} \n            Analyse the subgoal and its termination condition
2747         ↪ Instructions: \n            Analyse the subgoal and its termination condition
2748         ↪ and decide if the subgoal is completed. \n            Then use
2749         ↪ action-observation sequence: {action_sequence} to give a high level summary of current
2750         ↪ evaluation. This summary will later be passed to low level agent in order to improve its
2751         ↪ actions. \n            Remember that your summary will be passed to low level
2752         ↪ component in order to improve its actions. \n"
2753 ],
2754     "summariser": [
2755         1,
2756         "You are a critic module analyzing an agent's attempt to achieve a subgoal in a survival game
2757         ↪ environment where resources decay over time. Your task is to identify the **single most
2758         ↪ important factor** that caused SUCCESS or FAILURE by **comparing the current
2759         ↪ action-observation sequence to similar successful and failed attempts.**\n            Information:
2760         ↪ Target subgoal: {subgoal}\n            Outcome of the action-observation sequence: {outcome}\n            Action-observation history: {action_obs_seq}\n            Game context: {game_info}\n            Survival plan:
2761         ↪ {survival_plan}\n            Similar successful attempts: {entries_successful_goal}\n            Similar failed
2762         ↪ attempts: {entries_failed_goal}\n            Instructions: \n            Analyze the current {action_obs_seq} in
2763         ↪ the context of {entries_successful_goal} and {entries_failed_goal}. Focus on identifying key
2764         ↪ differences in resource management, timing, and actions taken.\n            Consider the resources
2765         ↪ available and their decay rates as indicated in {game_info}, paying close attention to how
2766         ↪ resource states differ between the successful, failed, and current attempt **at the moment
2767         ↪ of subgoal completion or failure.**\n            Identify the **single most critical divergence** that
2768         ↪ explains the outcome. This could be a specific resource that was more abundant (or less
2769         ↪ abundant) in the successful attempt, a crucial action that was taken (or not taken), or a
2770         ↪ timing difference that impacted resource availability.\n            Express the result in **one short
2771         ↪ sentence** highlighting the comparative aspect. For example: \"Unlike successful attempts,
2772         ↪ the agent failed to prioritize gathering berries before attempting to craft the tool,
2773         ↪ leading to starvation.\"\n            Or, \"The agent successfully gathered wood within the same
2774         ↪ timeframe as past successful attempts, but, unlike those attempts, the observation sequence
2775         ↪ shows the agent prioritized building a fire and not water collection which lead to
2776         ↪ dehydration and subsequent death.\"\n            If resource decay is not the primary factor revealed
2777         ↪ by the comparison, state the next most relevant factor based on the differences observed
2778         ↪ between the current attempt and {entries_successful_goal} and {entries_failed_goal}, also
2779         ↪ taking into account {survival_plan} and {obs}."
2780 ],
2781     "explorer": [
2782         0,
2783         "\n            Task: Create an exploration plan to help the agent discover new skills.\n\n            Data:
2784             - Game info: {game_info}\n             - Observation: {obs}\n             - Subgoal
2785             - summary: {summary}\n             - Recent 16 action\u00d7observation pairs: {history}\n"
2786 ]

```

```
2754
2755     ↪ Steps:\n          1. Analyze the environment and agent\u2019s situation.\n          2. Propose a\n          ↪ focused exploration plan with clear purpose, conditions, and indicators.\n          "
2756     ],
2757     "sequential": [
2758         0,
2759         "You always have to output one of the above actions at a time and no other text. You always\n         ↪ have to output an action until the episode terminates."
2760     ],
2761     "fitness": 39.090909090909086,
2762     "id": "97c9c973-9f66-4391-a4c1-f8904921e95d",
2763     "_std_error": 4.904037803367701
2764 }
```

MiniHack Final Genome

```

2768 {
2769     "hierarchy": 0,
2770     "amygdala": 0,
2771     "loop_evaluator": 0,
2772     "explorer": 0,
2773     "summariser": 0,
2774     "time_decay": [
2775         0,
2776         0.01
2777     ],
2778     "cosine_cutoff": [
2779         0,
2780         0.05
2781     ],
2782     "epsilon": [
2783         0,
2784         0.01
2785     ],
2786     "temperature": 1.036744932065481,
2787     "prompts": {
2788         "high_level": [
2789             0,
2790             "\nYou are a strategic planner for a video game AI. Analyze the current game state and create
2791             ↪ achievable subgoals that advance toward the main objective.\n\nREQUIREMENTS:\n- Subgoals
2792             ↪ must be immediately achievable with current capabilities\n- Focus on next logical steps, not
2793             ↪ distant goals\n- Each subgoal should have clear success criteria\n\nCURRENT STATE:
2794             ↪ (obs)\nGAME INFO: {game_info}\n\nCreate a sequential plan with 2-3 subgoals."
2795         ],
2796         "low_level": [
2797             0,
2798             "\n You are an action executor in a video game AI system. Given a subgoal from the high-level
2799             ↪ planner, propose a sequence of actions to achieve it.\n\n#####
2800             ↪ CURRENT SUBGOAL: {subgoal}\n
2801             ↪ CURRENT STATE: {obs}\n             ↪ PREVIOUS ACTIONS: {action_sequence}\n\n
2802             ↪ Plan the
2803             ↪ full sequence of actions needed to complete the subgoal. Avoid repeating actions if
2804             ↪ observations don't change.\n             ↪ Avoid extra commentary outside the REASONING and ACTIONS
2805             ↪ list.\n"
2806     ],
2807     "amygdala": [
2808         0,
2809         "\n      Decide if survival mode should activate.\n\n      Observation: {obs}\n
2810         ↪ Survival plan: {survival_plan}\n\n      1. Check if observation meets any subtask
2811         ↪ prerequisites.\n      2. If several match, pick highest priority.\n"
2812     ],
2813     "loop": [
2814         0,
2815         "\n      Task: Decide if the agent is stuck in a loop.\n\n      Loop = repeating actions
2816         ↪ without meaningful progress toward the subgoal\n      (progress = closer to goal, new
2817         ↪ info, removing failed paths, or advancing game state).\n\n      Data:\n
2818         ↪ Observation: {obs}\n             ↪ Subgoal: {subgoal}\n             ↪ Termination condition:
2819             ↪ (success_condition)\n             ↪ Action\u2013observation history: {history}\n
2820             ↪ Steps:\n             ↪ 1. Check if enough steps have been taken to allow exploration.\n             ↪ 2.
2821             ↪ Look for repeated patterns without progress.\n             ↪ 3. If loop detected, identify cause:
2822             ↪ missing info, unknown prerequisite, or unexplored path.\n"
2823     ],
2824     "termination": [
2825         0,
2826         "\n      You are a termination evaluator for a video game AI. Check if the agent has
2827         ↪ completed its subgoal.\n\n      CURRENT STATE: {obs}\n             ↪ SUBGOAL: {subgoal}\n
2828             ↪ SUCCESS CONDITION: {success_condition}\n             ↪ RECENT ACTIONS:
2829             ↪ {action_sequence}\n\n      Compare the current state with the success condition to
2830             ↪ determine if the subgoal is complete. Provide feedback to help the agent improve.\n"
2831     ],
2832     "summariser": [
2833         0,
2834         "\n      You are a critic analyzing an agent's subgoal attempt. Identify the key factor that
2835         ↪ caused success or failure.\n\n      SUBGOAL: {subgoal}\n             ↪ OUTCOME: {outcome}\n

```

```

2808
2809     ↪ ACTION HISTORY: {action_obs_seq}\n\n      Focus on specific resources and quantities that
2810     ↪ mattered most. If no resources involved, identify the next most important factor.\n      "
2811 ],
2812 "explorer": [
2813     0,
2814     "\n      Task: Create an exploration plan to help the agent discover new skills.\n\n
2815     ↪ Data:\n      - Game info: {game_info}\n      - Observation: {obs}\n      - Subgoal
2816     ↪ summary: {summary}\n      - Recent 16 action\u2013observation pairs: {history}\n\n
2817     ↪ Steps:\n      1. Analyze the environment and agent\u2019s situation.\n      2. Propose a
2818     ↪ focused exploration plan with clear purpose, conditions, and indicators.\n\n      "
2819 ],
2820 "sequential": [
2821     1,
2822     "As an AI survival agent operating within a dynamic resource-scarce environment, your
2823     ↪ objective is to maximize long-term survivability. Prioritize actions that maintain vital
2824     ↪ resource levels while mitigating immediate threats. Given your current observation ({obs}),
2825     ↪ game information ({game_info}) including potential actions, and the history of your past 16
2826     ↪ action-observation pairs ({history}), evaluate the following:\n\n1. **Resource
2827     ↪ Assessment:** Determine current levels of critical resources (e.g., health, energy, food,
2828     ↪ water) as reflected in {obs}. Identify actions within {game_info} that deplete or replenish
2829     ↪ these resources. Consider the 'survival_plan' for guidance on sustainable resource
2830     ↪ management.\n\n2. **Threat Analysis:** Identify immediate dangers based on {obs}. Prioritize
2831     ↪ actions that avoid or neutralize these threats, considering the action-observation sequence
2832     ↪ towards the current subgoal ('{action_obs_seq}').\n\n3. **Goal Alignment:** Assess how each
2833     ↪ possible action aligns with your current subgoal ({subgoal}) and overarching survival plan
2834     ↪ ('{survival_plan}'). Use '{entries_successful_goal}' and '{entries_failed_goal}' to learn
2835     ↪ from past attempts to achieve similar subgoals.\n\n4. **Predictive Risk Mitigation:***
2836     ↪ Evaluate the potential for each action to lead to a critical failure within the next few
2837     ↪ steps. Prioritize actions that maintain options and avoid irreversible negative consequences
2838     ↪ based on your history ('{history}'). The 'success_condition' should also be
2839     ↪ considered.\n\nSelect the single most optimal action from {game_info} that balances resource
2840     ↪ acquisition/conservation, threat mitigation, goal progression, and predictive risk
2841     ↪ mitigation. Justify your selection briefly based on the above analysis.\n\nOutput format:
2842     ↪ ACTION: [selected action] | RATIONALE: [brief justification]\n\nYou must provide an output
2843     ↪ in this format at each step until the episode terminates. Do not output any other text. If
2844     ↪ no immediately safe or advantageous action is available, select the least detrimental action
2845     ↪ while adjusting your 'survival_plan' accordingly."
2846   ],
2847   "fitness": 22.5,
2848   "id": "d1914812-4881-4b8e-85f6-ee47ccce9f47",
2849   "std_error": 6.602556323122129
2850 }
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861

```

NetHack Final Genome

```

2836
2837
2838
2839   {
2840     "hierarchy": 1,
2841     "amygdala": 0,
2842     "loop_evaluator": 1,
2843     "explorer": 0,
2844     "summariser": 1,
2845     "time_decay": [
2846       1,
2847       0.004870010771374662
2848     ],
2849     "cosine_cutoff": [
2850       1,
2851       0.01
2852     ],
2853     "epsilon": [
2854       0,
2855       0.01
2856     ],
2857     "temperature": 1.0,
2858     "prompts": {
2859       "high_level": [
2860         1,
2861         "You are a strategic planner in a hierarchical video game AI system. Your role is to analyze
2862         ↪ the current game state and generate achievable subgoals that strategically advance toward
2863         ↪ the main objective.\n\n          CRITICAL CONSTRAINTS:\n          ↪ Subgoals must be immediately achievable given the current agent state and
2864          ↪ capabilities\n          - Focus on the next logical progression steps, not
2865          ↪ distant end-goals\n          - Each subgoal should have clear, observable
2866          ↪ success criteria\n          - Subgoals should be novel and interesting,
2867          ↪ compared to previous attempts\n          - **Subgoals should consider
2868          ↪ resource acquisition and conservation. Avoid actions that waste valuable resources unless
2869          ↪ absolutely necessary for survival or progression.\n\n          Below I
2870          ↪ will provide the game description, available actions, and current state information.\n          Game description:\n          {game_info}\n          #####\n          CURRENT CONTEXT: \n\n          - Game State: {obs} \n\n
2871          ↪ - Survival plan provided by the survival planner, that you should
2872          ↪ consider for your tasks: {survival_plan}\n          - Here are most
2873          ↪ similar successful entries from the archive: {entries_successful_goal} \n\n
2874       ],
2875     }
2876   }
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
3999

```

```

2862
2863     - Here are most similar failed entries from the archive: {entries_failed_goal}
2864     \n\n
2865     ANALYSIS FRAPAPMEWORK: 1.
2866     Analyse the summary from previous runs and let it guide your decision making. \n
2867     2. Assess what is immediately possible given current agent state and
2868     environment\n
2869     3. Identify what kind of actions could be
2870     considered novel or interesting\n
2871     4. Identify the most direct
2872     path toward the main objective\n
2873     5. Select subgoals that form a
2874     logical sequence\n
2875     6. Ensure each subgoal can be verified through
2876     observable game state changes\n\n
2877     - Feasible: Can be started immediately with current
2878     resources/position\n
2879     - Measurable: Success/failure can be
2880     determined from game observations\n
2881     - Progressive: Each subgoal
2882     enables the next or advances toward main goal\n
2883     - Specific: Clear
2884     enough for a lower-level agent to understand and execute\n
2885     - Considerate of the summary of previous runs\n\n
2886     Make sure that
2887     your subgoals are sequential."
2888 ],
2889 "low_level": [
2890     1,
2891     "You are an important executor component of a hierarchical video game system. You are given
2892     one of higher level option and its termination condition proposed by the higher level
2893     planner. Your role is to propose a sequence of actions that will make you progress towards
2894     the given option.\"
2895     Below I will provide you with the game
2896     description, possible actions you can take and the overall goal of the game.\n\n
2897     #####\n
2898     Here is a subgoal provided by the high level planner that you should focus on completing:
2899     {subgoal} \n\n
2900     Here is your current state: {obs} \n\n
2901     Here is the action-observation sequence towards current subgoal:
2902     {action_sequence} \n\n
2903     Here are the most similar successful
2904     entries from the archive: {entries_successful_goal} \n\n
2905     Here are
2906     the most similar failed entries from the archive: {entries_failed_goal} \n\n
2907     Use the action and observation sequence together with the current state to decide
2908     the **full ordered sequence of actions** that will achieve the subgoal. \n\n
2909     Avoid repeating the same actions if the observation doesn't change. \n"
2910 ],
2911 "amygdala": [
2912     0,
2913     "\n      Decide if survival mode should activate.\n\n      Observation: {obs}\n
2914     \n      Survival plan: {survival_plan}\n\n      1. Check if observation meets any subtask
2915     \n      prerequisites.\n      2. If several match, pick highest priority.\n"
2916 ],
2917 "loop": [
2918     1,
2919     "You are a Senior Game AI Debugging Specialist, tasked with analyzing the behavior of an agent
2920     in a hierarchical video game system. Your primary goal is to determine if the agent's
2921     current behavior constitutes a genuine \"stuck\" state, which requires
2922     intervention.\n\nConsider the following details about the agent's progress:\n\n* **Current
2923     Observation:** {obs} * **Current Subgoal:** {subgoal} * **Success Condition:** *
2924     {success_condition}\n* **Action-Observation History:** {history} * **Game
2925     Information:** {game_info} * **Survival Plan:** {survival_plan}\n\nCritically evaluate
2926     the agent's actions, considering the following factors specific to Game AI:\n\n1.
2927     **Information Gain vs. Redundancy:** Is the agent genuinely gathering *new* and *relevant*
2928     information? Merely observing a change in the environment after an action doesn't
2929     necessarily indicate progress. Determine if the information gained helps reduce uncertainty
2930     related to achieving the current subgoal or aligns with the overall game objective, given by
2931     'survival_plan'.\n2. **Exploration Strategy:** Is the agent's exploration strategy
2932     sufficient to overcome potential local minima or deceptive landscapes? Many \"stuck\"
2933     situations arise from poor exploration. Consider if the agent's 'action_sequence' explores
2934     diverse enough actions, or if it is repeating the same action variations in a small
2935     area.\n3. **World Model Limitations:** Assess whether the agent's internal model of the
2936     game world (implied by its actions and reactions in 'action_obs_seq' and 'history') is
2937     incomplete or inaccurate. Is it missing key information about the game mechanics, object
2938     interactions, or environmental dynamics? If so, suggest avenues for improving the world
2939     model, such as targeted exploration or observation of specific events.\n4. **Leverage
2940     entries_successful_goal and entries_failed_goal:** Look into similar subgoals in the past to
2941     help you understand how the agent behaved then and compare that with the current
2942     behavior.\n\n**Your Task:**\n\nBased on the provided details and your expertise in Game AI,
2943     determine if the agent is genuinely stuck, meaning it's unlikely to achieve its subgoal
2944     without external intervention. Focus on *why* the agent is stuck. Specifically, is the
2945     agent's failure due to:\n\n* A lack of crucial information that could be obtained through
2946     more effective exploration?\n* An inaccurate or incomplete world model preventing it from
2947     making informed decisions?\n* A fundamental flaw in its action selection
2948     strategy?\n\nProvide a concise justification for your conclusion, outlining the specific
2949     factors that support your assessment. Prioritize identifying concrete steps the agent could
2950     take to overcome the \"stuck\" state, considering the limited information it may possess.
2951     Avoid vague statements and focus on actionable recommendations rooted in Game AI best
2952     practices."
2953 ],
2954 "termination": [
2955     1,
2956     "You are an important termination evaluator component of a hierarchical video game system,
2957     functioning as a specialized AI reinforcement learning analyst.\n\nYour task is to:\n\n1.
2958     Determine whether the agent has met the termination condition for a subgoal.\n\n2. Provide a
2959     concise summary that will help guide the lower-level agent's future actions, specifically
2960     addressing potential issues related to reinforcement learning
2961     strategies.\n\nDetails:\n\nHere is your current state that you should compare with the
2962     termination condition: {obs}\n\nHere is the subgoal the lower-level agent is working
2963     towards: {subgoal}\n\nHere is the termination condition of the above subgoal given by the
2964     higher-level agent: {success_condition}\n\nInstructions:\n\nAnalyze the subgoal and its
2965     termination condition and decide if the subgoal is completed. Then, using the

```

```

2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969

    ↪ action-observation sequence: {action_obs_seq}, provide a high-level summary of your current
    ↪ evaluation.\n\nIn your summary, consider the following reinforcement learning concepts:\n\n*
    ↪ **Exploration vs. Exploitation:** Is the agent adequately exploring the environment, or is
    ↪ it prematurely exploiting a sub-optimal strategy? If the agent has been repeating the same
    ↪ actions in similar states (as observed in {history}) without success, encourage more
    ↪ exploration.\n\n* **Reward Shaping:** Consider how the higher-level agent's reward
    ↪ structure (implicit or explicit) might be influencing the agent's behavior. Is the agent
    ↪ being incentivized towards unintended consequences?\n\n* **Local Optima:** Is the agent
    ↪ potentially stuck in a local optimum? Compare the current situation with
    ↪ {entries_failed_goal} to identify if similar action sequences have previously led to
    ↪ failure. If so, suggest a drastic change in strategy.\n\n* **Credit Assignment:** How can
    ↪ the agent better understand which actions are contributing to success or failure? Suggest
    ↪ focusing on the most recent actions, especially in light of {observation_sequence}.\n\nIf
    ↪ the agent has failed similar subgoals in the past, provide an alternative approach based on
    ↪ {entries_successful_goal}. Highlight alternative actions it can take in a similar
    ↪ situation.\n\nRemember that your summary will be passed to the low-level component to
    ↪ improve its actions. It should be actionable and specific. It should be in simple language
    ↪ for the low-level agent to understand and must contain suggestions to avoid common RL
    ↪ pitfalls based on the information provided. The survival plan is {survival_plan} for high
    ↪ level goals."
],
"summariser": [
    1,
    "You are a critic module analyzing an agent's attempt to achieve a subgoal in a game
    ↪ environment, considering the agent's survival plan.\n\nYour task is to identify the **single
    ↪ most important factor** that caused SUCCESS or FAILURE, given the broader context of
    ↪ survival.\n\nInformation:\n- Target subgoal: {subgoal}\n- Outcome of the action-observation
    ↪ sequence: {outcome}\n- Action-observation history: {action_obs_seq}\n- Game context:
    ↪ {game_info}\n- Agent's Survival Plan: {survival_plan}\n- Success Condition:
    ↪ {success_condition}\n\nInstructions:\n- First, read the survival plan and understand the key
    ↪ threats and resource priorities.\n- Then, analyze the action-observation sequence in the
    ↪ context of the target subgoal and survival plan.\n- Focus on **specific resources and their
    ↪ quantities** that were critical according to the survival plan. Consider if failing to meet
    ↪ {success_condition} resulted from a resource shortage, specifically referencing the
    ↪ quantities mentioned in the action-observation history.\n- How did the agent's actions
    ↪ either help or hinder the broader survival strategy defined in {survival_plan}?\n- If
    ↪ resources were missing that were crucial to survival, state **exactly which resources and
    ↪ how many** were missing and how it violated the survival plan.\n- If no resources are the
    ↪ primary issue, state the next most relevant factor that impacted both the subgoal and
    ↪ survival chance.\n- Express the result in **one short sentence** highlighting the connection
    ↪ to {survival_plan}."
],
"explorer": [
    0,
    "\n        Task: Create an exploration plan to help the agent discover new skills.\n\n
    ↪ Data:\n        - Game info: {game_info}\n        - Observation: {obs}\n        - Subgoal
    ↪ summary: {summary}\n        - Recent 16 action\observation pairs: {history}\n\n
    ↪ Steps:\n        1. Analyze the environment and agent's situation.\n        2. Propose a
    ↪ focused exploration plan with clear purpose, conditions, and indicators.\n\n"
],
"sequential": [
    0,
    "You always have to output one of the above actions at a time and no other text. You always
    ↪ have to output an action until the episode terminates."
]
},
"fitness": 0.8527708222454596,
"id": "aac4dfb-4a3c-4090-a8bc-5f9265d65eda",
"_std_error": 0.4739428639198163
}

```