

# Justificativa para a Modelagem Condicional da Função de Geração de Conteúdo Gamma

Ismael S. Silva

8 de julho de 2025

## Abstract

Este documento apresenta a justificativa e a arquitetura formal da função de geração procedural de conteúdo, denominada Gamma ( $\Gamma$ ), para o ambiente de simulação multiagente *BuriedBrains*. Argumentamos que, para testar rigorosamente hipóteses sobre comportamentos emergentes e sensibilidade ao risco, uma geração puramente aleatória é insuficiente, pois introduz ruído estocástico que obscurece os padrões de aprendizado. Em contrapartida, um modelo excessivamente determinístico falha em promover a generalização de políticas. Propomos uma arquitetura de geração sequencial e condicional que modela as interdependências entre os elementos do ambiente. A função  $\Gamma$  utiliza um sistema de orçamento de complexidade, sensível ao contexto externo (trajetória do agente  $k$  e topologia  $T$ ), para guiar uma cadeia de decisões probabilísticas que possuem coerência interna. O resultado é a criação de uma paisagem de risco e oportunidade que é ao mesmo tempo dinâmica, sistêmica e rica em contexto, constituindo um componente fundamental do design experimental deste benchmark.

## 1 Introdução e Motivação

O ambiente *BuriedBrains* foi concebido como um laboratório computacional para o estudo de comportamentos complexos em agentes de inteligência artificial com memória. Um pilar central deste ambiente é a sua natureza procedural, que visa testar a capacidade de generalização e planejamento de longo prazo dos agentes. Para que este objetivo seja alcançado, a função de geração de conteúdo,  $\Gamma$ , não pode ser um mero mecanismo de aleatorização. Ela deve ser um componente integrante do design experimental.

A motivação para uma modelagem mais sofisticada surge da insuficiência de abordagens mais simples para a criação de um benchmark robusto:

- **Geração Puramente Aleatória:** Um modelo que seleciona conteúdo de forma arbitrária falha em criar um gradiente de complexidade coerente. O risco se torna uniforme e imprevisível, tornando impossível testar hipóteses sobre estratégias sensíveis ao risco (H2), pois o agente não tem uma estrutura de risco para aprender.
- **Geração Estruturada Rígida:** Um modelo que segue templates fixos, embora controlado, torna-se previsível. Isso compromete a necessidade de generalização (H4), pois o agente pode simplesmente memorizar sequências de estados ótimos, em vez de aprender as mecânicas subjacentes do ambiente.

Diante disso, a função  $\Gamma$  foi projetada para atuar como um mecanismo de orquestração procedural, que instancia desafios contextuais e sistemicamente coerentes, em vez de obstáculos aleatórios.

## 2 Arquitetura da Função Gamma

A arquitetura escolhida para  $\Gamma$  é sequencial, condicional e baseada em um sistema de economia interna (custo e orçamento). Ela decompõe a complexa tarefa de instanciar o conteúdo de um vértice em uma cadeia de decisões probabilísticas interdependentes, que garantem coerência lógica e estratégica ao desafio gerado.

### 2.1 Estrutura Baseada em Slots e Categorias

Para garantir controle estrutural, o vetor de atributos de um vértice,  $\alpha(v)$ , é definido como uma tupla de três elementos, onde cada elemento ("slot") corresponde a uma categoria de conteúdo:

$$\alpha(v) = (c_i, c_e, c_f)$$

Onde  $c_i \in \text{Pool}_{\text{inimigos}} \cup \{\emptyset\}$ ,  $c_e \in \text{Pool}_{\text{eventos}} \cup \{\emptyset\}$ , e  $c_f \in \text{Pool}_{\text{efeitos}} \cup \{\emptyset\}$ . Esta estrutura impõe um limite máximo de um item por categoria, evitando a geração de configurações caóticas e não balanceadas.

### 2.2 O Sistema de Orçamento e Custo

O núcleo do mecanismo de controle é um sistema de economia de complexidade.

- **Custo:** A cada entidade de conteúdo  $j$  é associado um  $\text{Custo}(j) \in \mathbb{R}$ , que quantifica seu impacto na complexidade do desafio.
- **Orçamento:** A cada vértice  $v$  é alocado um  $\text{Orçamento}(v) \in \mathbb{R}^+$ , que representa sua capacidade total de desafio.

Este sistema garante que desafios de alto custo (e.g., 'Inimigo de Elite') só possam ser instanciados em vértices com um orçamento correspondentemente alto.

### 2.3 Modelo de Geração Condicional e Sequencial

A principal característica do modelo é que a geração de conteúdo não ocorre de forma paralela, mas como uma cadeia de eventos condicionais. A geração é guiada por um contexto **externo** (topologia e progresso) e um contexto **interno** (as entidades já alocadas nos slots do próprio vértice).

Primeiramente, o orçamento inicial,  $B_0$ , é determinado pelo contexto externo. Seja  $k$  a profundidade do agente na trajetória e  $T$  o vetor de informação topológica, temos:

$$B_0 = B(k, T)$$

Isso garante que vértices em estágios avançados ou em posições estratégicas recebam um orçamento maior. A partir daí, a geração segue uma sequência que modela a distribuição de probabilidade conjunta dos atributos como:

$$P(c_i, c_e, c_f \mid B_0) = P(c_i \mid B_0) \cdot P(c_e \mid c_i, B_0) \cdot P(c_f \mid c_i, c_e, B_0)$$

Na prática, o processo utiliza um orçamento residual para restringir as escolhas subsequentes:

1. **Geração do Inimigo ( $c_i$ ):** O primeiro slot é preenchido com base no orçamento inicial.

$$c_i \sim P(\cdot \mid B_0)$$

2. **Geração do Evento ( $c_e$ ):** O orçamento é atualizado, e o evento é escolhido com base no inimigo já presente e no orçamento restante.

$$B_{\text{res}_1} = B_0 - \text{Custo}(c_i)$$

$$c_e \sim P(\cdot \mid c_i, B_{\text{res}_1})$$

3. **Geração do Efeito ( $c_f$ ):** O processo se repete, condicionando a escolha final aos elementos já definidos e ao orçamento final.

$$B_{\text{res}_2} = B_{\text{res}_1} - \text{Custo}(c_e)$$

$$c_f \sim P(\cdot \mid c_i, c_e, B_{\text{res}_2})$$

Este modelo sequencial garante que a configuração final do vértice seja sistemicamente coerente. Por exemplo, a presença de um ‘Inimigo Forte’ pode aumentar a probabilidade de um ‘Terreno Lento’, criando um desafio sinérgico e logicamente consistente.

### 3 Vantagens e Implicações para a Pesquisa

A adoção desta arquitetura para a função  $\Gamma$  oferece vantagens cruciais para os objetivos do *BuriedBrains* como plataforma de benchmark.

- **Criação de Paisagens de Risco Sinérgicas:** Ao modelar as interdependências entre os desafios, o ambiente apresenta ao agente paisagens de risco com coerência interna. O perigo não é apenas a soma das partes, mas o resultado de suas sinergias. Isso é essencial para investigar se o agente desenvolve políticas prudentes (H2), capazes de avaliar riscos multifacetados e contextuais.
- **Promoção de Generalização Profunda:** O ambiente exhibe padrões contextuais consistentes e sistêmicos. Para obter um desempenho ótimo, o agente é forçado a aprender um modelo implícito do mundo, inferindo as "regras ocultas" que governam as correlações entre as entidades (e.g.,  $P(c_f|c_i)$ ). Isso cria um teste muito mais robusto para a generalização de políticas (H4), indo além do reconhecimento de padrões superficiais.
- **Controle e Reprodutibilidade:** A natureza parametrizada do sistema (custos, pesos, probabilidades condicionais) permite um balanceamento fino e a condução de experimentos controlados e reprodutíveis, que são a base de qualquer investigação científica rigorosa.

## 4 Conclusão

A modelagem da função  $\Gamma$  como um processo de geração sequencial e condicional é uma decisão de design deliberada para alinhar o ambiente *BuriedBrains* aos seus objetivos de pesquisa. Esta abordagem substitui o ruído de uma aleatoriedade pura pela complexidade rica de um sistema com coerência interna e sinergias estratégicas. Ao fazê-lo,  $\Gamma$  transforma o ambiente em uma plataforma de benchmark rigorosa, projetada especificamente para medir e analisar a emergência de estratégias inteligentes que devem compreender e explorar as regras subjacentes de um mundo dinâmico e parcialmente observável.