

EvoGen Framework 設計仕様書

フレームワーク設計仕様書

1. フレームワークの名称

- **EvoGen Framework** (Evolutionary Generative Framework)
- **Synaptic Ascent Framework**
- **IDEAs** (Iterative Design Evolution via AI Synthesis)

ここでは、**EvoGen Framework** を正式名称として進めます。

2. 目的と概要

目的:

本フレームワークは、人間では思いつきにくい、または見落としがちな高品質な解決策を、LLM（大規模言語モデル）を活用して自律的かつ進化的に探求することを目的とします。

概要：ユーザーが定義した課題に対し、LLMが複数の初期解を生成します。次に、別のLLMがそれらの解を多角的に評価し、スコアリングします。その評価結果に基づき、優れた解の要素を組み合わせ（交叉）、新たな視点を加える（突然変異）ことで、次世代のより洗練された解を生成します。この「生成 → 評価 → 進化」のサイクルを繰り返すことで、解の質を継続的に向上させ、最終的に最適な解決案へと収束させます。

3. 構成要素

フレームワークは、それぞれ異なる役割を持つ3つのAIエージェントによって構成されます。

- ① **Creative Genesis Agent** (創造的生成エージェント):
 - 役割: 課題定義に基づき、多様性に富んだ初期解決案を複数生成するブレイнстーマー。常識にとらわれない斬新なアイデアを出すことを得意とします。
- ② **Multi-faceted Scrutiny Agent** (多角的評価エージェント):
 - 役割: 生成された解決案を、ユーザーが定めた評価基準（実現可能性、コスト、革新性など）に従って、客観的かつ厳密に評価・採点する批評家。各評価の根拠も明確に言語化します。
- ③ **Evolutionary Synthesis Agent** (進化的統合エージェント):
 - 役割: 高評価の解決案から優れた遺伝子（アイデアの核）を抽出し、それを巧みに組み合わせたり、新たな変異を加えたりすることで、次世代のより強力な解決案を生み出す戦略家。

4. 実行手順（ステップ・バイ・ステップ）

ステップ 0: 課題定義

- **目的:** AI が解くべき問題を正確に理解するための初期設定。
- **入力:** ユーザーからの指示。具体的には以下の要素を含む。

課題文 : 解決したい問題の内容。

要件・制約条件: 解決案が満たすべき必須条件やリソース上の制約。

評価軸 : 何を重視するか（例:「コストを最重視し、次に実現可能性を重視する」）。

- **処理内容:** ユーザーは指定されたフォーマットで課題を入力します。
- **出力:** 全エージェントが共有する、構造化された「課題定義書」。

ステップ 1: 初期解の生成 (Generation 0)

- **目的:** 探索の出発点となる、多様な解決案群の作成。
- **入力:** ステップ 0 で作成された「課題定義書」。

処理内容: **Creative Genesis Agent** が課題定義書を解釈し、指定された数（例: 5 つ）の多様な初期解決案を生成します。

出力 : 5 つの初期解決案のリスト。

ステップ 2: 解の評価

- **目的:** 生成された各解決案の品質を客観的な指標で測定。

入力 : 解決案のリストと「課題定義書」。

処理内容: **Multi-faceted Scrutiny Agent** が、各解決案を「課題定義書」の評価軸に基づき個別に評価します。評価はスコア（例: 100 点満点）と、そのスコアに至った論理的な根拠のテキストで構成されます。

出力: 各解決案にスコアと評価理由が付与された「評価済み解決案リスト」。

ステップ 3: 解の選択と淘汰

- **目的:** 次世代を生み出すための「親」となる優れた解を選び出す。
- **入力:** 「評価済み解決案リスト」。
- **処理内容:** 評価スコアに基づき、上位の解決案（例: 上位 3 案）を選択します（**エリート選択**）。これにより、優れた特性が次世代に確実に引き継がれます。
- **出力:** 次世代生成の元となる「選択済み解決案リスト」。

ステップ 4: 次世代解の生成 (Generation N+1)

- **目的:** 優れた親の特性を受け継ぎつつ、さらに改良された新しい解を生み出す。

入力：「選択済み解決案リスト」と「評価済み解決案リスト」全体。

- **処理内容:** Evolutionary Synthesis Agent が以下の処理を行います。
 1. **交叉:** 選択された複数の解から優れた部分（例: A 案のコスト削減アイデアと、B 案の画期的な UI）を組み合わせて新しい解を作成します。
 2. **突然変異:** 高評価の解に、あえて新しい視点やランダムな要素を加え、局所最適解に陥るのを防ぎます。
 3. **新規生成:** スコアが低かった解も参考にし、「なぜ評価が低かったのか」を反面教師として全く新しい解を 1 つ生成させ、多様性を維持します。

出力：新たに生成された 5 つの次世代解決案のリスト。

ステップ 5: 繰り返しと収束

- **目的:** 解の品質が十分に高まるまで、進化のサイクルを継続する。
- **入力:** 次世代解決案のリスト。

処理内容: ステップ 2（評価）からステップ 4（次世代生成）のサイクルを、あらかじめ定められた条件を満たすまで繰り返します。

○ 終了条件の例:

- 指定された世代数（例: 5 世代）に達した時。
- 世代間の最高スコアの向上が一定値以下（例: 3%未満）になった時（収束）。
- **出力:** 繰り返す場合は次のサイクルのための解決案リスト。終了する場合は最終世代の「評価済み解決案リスト」。

ステップ 6: 最終解の提示

- **目的:** 進化のプロセスを経て得られた最も優れた解をユーザーに提示する。
- **入力:** 最終世代の「評価済み解決案リスト」。
- **処理内容:** 全世代を通じて最も評価の高かった解決案を「最適解」として選択します。また、それに次ぐ高評価の案もいくつか「次善策」として提示します。
- **出力:** 最適解、次善策、そしてそこに至るまでの進化の過程をまとめた最終レポート。

5. 各ステップのプロンプト設計指針

初期解作成プロンプトの指針 (for Creative Genesis Agent)

- **役割の明確化:** あなたは一流のイノベーターです。常識にとらわれず、大胆でユニークなアイデアを出すことを期待されています。
- **多様性の強調:** 以下の課題に対し、互いに全く異なるアプローチからの解決策を 5 つ提案してください。例えば、技術的な解決策、組織的な解決策、心理的なアプロ

ーチなど、多角的な視点から考えてください。

- **構造化出力の要求:** 各解決策は「名称」「概要」「具体的な方法」の 3 つの項目で記述してください。

解評価プロンプトの指針 (for Multi-faceted Scrutiny Agent)

- **役割の明確化:** あなたは客観的で厳しい批評家です。感情や先入観を排除し、与えられた評価基準にのみ基づいて冷静に分析してください。
- **評価基準の明示:** 以下の解決案を、下記の評価基準で 100 点満点で採点してください。
 - 実現可能性: 40 点
 - 革新性: 30 点
 - コスト効率: 30 点
- **構造化出力の要求:** 評価結果は必ず以下の JSON 形式で出力してください。

JSON

```
{  
    "solution_name": "評価対象の解決案名",  
    "scores": {  
        "feasibility": 0,  
        "innovation": 0,  
        "cost_efficiency": 0  
    },  
    "total_score": 0,  
    "strengths": "この案の優れている点",  
    "weaknesses": "この案の懸念点や改善点",  
    "overall_comment": "総評"  
}
```

次世代解作成プロンプトの指針 (for Evolutionary Synthesis Agent)

- **役割の明確化:** あなたは優れた戦略家であり編集者です。与えられた素材（既存の解決案とその評価）の長所を組み合わせ、短所を克服する、より優れた新しい案を合成してください。
- **具体的な指示:**
 - **交叉:** 以下の高評価の解決案リスト A、B、C から、それぞれの最も優れた要素を抽出・融合し、全く新しい解決案を 2 つ生成してください。
 - **突然変異:** 最も評価の高かった解決案 A をベースに、あえて一つの制約条件を無視した場合、または全く新しい技術（例：量子コンピューティング）を取り入れた場合どうなるか、という思考実験を行い、飛躍のある解決案を 2 つ生成してください。

- **反面教師:** 低評価だった解決案 D の「評価が低かった理由」を分析し、その失敗を完全に克服する新しいアプローチの解決案を 1 つ生成してください。
-

6. フレームワークの利点と考慮点

利点:

- **アイデアの自動拡張:** 人間の思考の癖やバイアスから解放され、多様で斬新なアイデアを網羅的に探求できます。
- **品質の継続的向上:** 世代を重ねるごとに客観的な評価に基づいて解が改良されるため、品質の向上が保証されます。
- **プロセスの自動化:** アイデア出しから評価、改善までの一連のプロセスを自動化し、人間の創造的作業を支援・高速化します。
- **意思決定の支援:** 最終的に提示される解には、評価の根拠が明確に示されているため、人間が意思決定を行う際の強力な判断材料となります。

考慮点:

- **コスト:** LLM の API を繰り返し呼び出すため、計算コストが高くなる可能性があります。
 - **局所最適解への陥り:** 突然変異の仕組みが弱いと、初期のアイデアの範囲内での改善に留まり、画期的な解決策に到達できない可能性があります。
 - **評価基準の重要性:** フレームワークの性能は、最初に設定する「評価軸」の質に大きく依存します。評価軸が不適切だと、望ましくない方向に進化が進む可能性があります。
-

7. 可視化（テキストベースのフローチャート）

[ユーザー]

|

V

[ステップ 0: 課題定義書を作成]

|

+-----+
|

|

V (課題定義書)

| (繰り返し: 3~5 回)

[ステップ 1: Creative Genesis Agent が初期解(5 つ)を生成]

|

V (5 つの解)

|

[ステップ 2: Scrutiny Agent が各解を評価・スコアリング]

|

|

