

芦屋大学論叢 第84号
(令和7年7月30日)抜刷

小中学生向け生成 AI 教育支援システムの 初期実装における課題と設計的考察

小 澤 雄 生
安 東 茂 樹

小中学生向け生成 AI 教育支援システムの 初期実装における課題と設計的考察

小 澤 雄 生 (1)

安 東 茂 樹 (2)

(1) 京都教育大学附属京都小中学校

(2) 芦屋大学大学院特任教授

1. はじめに

近年、ChatGPT や Copilot などに代表される生成 AI (Generative AI) は、教育現場にも急速に広まりつつある。中でも今後使用が想定される比較的若年層の学習者にとって、生成 AI の活用は思考補助や表現支援といったポジティブな側面を持ちながらも、誤情報の生成、依存傾向、情報モラルの未成熟といったリスクも併せ持つと考えられる。とくに、プロンプトの設計や、生成された出力の批判的吟味といった情報活用能力は、十分に育成されていない段階の中学生にとっては高いハードルとなる可能性がある。

こうした状況に対して、教育現場では生成 AI の活用に向けたルール整備や活用指針が求められているが、学習者自身が安全かつ主体的に生成 AI を活用するための支援ツールや教育システムの開発は、十分に進んでいないのが現状である。

一方で、初等教育段階における AI リテラシー育成を目指した教材開発や授業実践の試みも見られる。たとえば、松田ら (2021)¹⁾ は、STEM/STEAM 教育の観点から小学生を対象に、AI 実装の仕組みや IoT 社会への理解、AI 技術への興味・関心を高める教材を開発し、試行授業を通じてその有効性を評価している。この研究では、児童が AI と自然に触れ合うことで、AI を身近に感じ、データと学習の関係性や AI との共生のイメージを体感的に学ぶことができたとされている。しかしながら、こうした取り組みの多くは AI 技術の基本理解や興味喚起にとどまっており、生成 AI のような高度なアウトプット型 AI との対話や、倫理的・批判的に利用するスキルの育成にまで踏み込んだ支援環境は、現状では十分といえない。

本研究では、中学生が生成 AI を「安全に」「教育的に」活用するための支援的な活用環境の整備を目的とし、そのプロトタイプの初期実装を試みた。その中で、Google Colaboratory (以下、Colab) 上で Gemini API を活用する構成を採用したが、実装段階では API 認証、モデル選択、ドキュメントの不整合など複数の技術的課題に直面した。

本研究では、そうした初期実装における技術的試行の記録と分析を通じて、教育用生成 AI システムの開発における課題と改善の視点を明らかにすることを目的とする。

2. 研究の背景と目的

2.1 初等中等教育における生成 AI 活用とその課題

生成 AI 技術は、教育現場にも急速に導入されつつあり、特に中学生を含む初等中等教育段階における利活用が注目されている。しかし、生成 AI は便利な道具である一方で、誤情報の生成、思考の依存、倫理的判断の難しさなど、学習者にとって大きなリスクも内在している。文部科学省（2024）²⁾ が公表した「初等中等教育段階における生成 AI の利活用に関するガイドライン」においても、学習者の年齢や発達段階に応じた段階的・限定的な活用が強く求められており、生成 AI の出力を鵜呑みにせず、思考と対話を通じて活用できるよう設計が必要であることが指摘されている。

また、厚生労働省（2022）³⁾ の「生成 AI の技術動向と影響」では、生成 AI の進展が教育・労働など多方面に与える影響とともに、それに対応するための情報リテラシーや支援体制の整備が重要であると指摘されている。そのため、若年層が生成 AI を活用する際には、単なる技術の導入にとどまらず、使用状況を見守り、適切なフィードバックや制御が可能なシステムの設計が求められる。

2.2 本研究の目的と位置づけ

本研究は、筆者が進めている研究「中学生の安全な生成 AI 利用支援のための教育用プロトタイプシステム設計」の一環として行ったものである。中学生が AI と安全に関わり、思考や表現を深められるような支援のあり方を探ることが主な目的である。そのためには、学習者の発達段階に応じて、生成 AI の活用を適切に支援・調整できるような設計が必要であり、たとえば AI の動作原理や限界を理解できるインタフェース、使用状況を把握して支援できる仕組みなど、多面的な環境整備が求められる。

本論文ではその第一歩として、Colab 上で Gemini API を用いたプロトタイプの初期実装を試みた。実装過程では、API 仕様の理解不足による認証形式の誤解、利用可能モデルの確認不足、ライブラリのバージョン管理の不整合といった複数の課題に直面し、安定した動作環境の構築には困難を伴った。これらの課題の一部には、API 技術そのものというよりも、利用環境（Colab）特有の構成や操作に起因する側面も含まれていた。

そこで本研究では、その失敗の過程を記録・分析し、今後の教育支援システム設計に反映する知見を明らかにすることを目的とする。これにより、同様のプロトタイプ構築を行う教育実践者や研究者に対し、有用な教訓と実装指針が提供できると考える。

また、本研究におけるプロトタイプ実装には、Python 言語と Colab を用いた。Python は、主要な生成 AI API（Gemini、ChatGPT など）の公式サポート言語であり、教育現場においても中学校・高等学校の情報教育に広く導入されつつある。筆者自身も、日常的に Python によるプログラミング教育を実践しており、Colab を活用した授業設計を行っている。Colab はインストール不要でクラウド上で動作し、Google アカウントを通じて簡単にアクセス・共有が可能であるため、教育現場との親和性が高い。

このように、Colab と Python の組み合わせは、学習者が AI と対話的に関わりながら、試行錯誤を通じて学びを深める学習環境を提供する点で、本研究の目的に極めて適した技術選択と推察される。さらに、教師側が教材を共有・管理しやすいという点でも、将来的に支援者による運用支援や学習履歴の活用を想定した拡張にも柔軟に対応可能な基盤であるといえる。

3. 開発方法と技術選定

3.1 システム設計の基本方針

本研究は、中学生が生成 AI を安全かつ段階的に活用できるよう支援する教育システムの構築を目的としている。特に、生成 AI との対話を通じて思考を深め、表現力を育むための環境整備を行う中で、「誤情報や依存への対処」「倫理的判断の支援」「教員によるガイドや制御の導入可能性」といった観点が重要となる。

この目的を実現するために、プロトタイプシステムの構成は以下のような要件を前提とした。

- (1) 生徒の入力（プロンプト）に対し、生成 AI が応答を返す構成とする。
- (2) 指導者が内容や履歴を確認できる「教師モード」の導入を視野に入れる。
- (3) プロンプトや出力のログを保存し、学習の振り返りや使用状況の記録ができるようにする。
- (4) 教員やシステムによるキーワードベースのフィルタリングなど、セーフティ機能の拡張を可能にする。

こうした要件に基づき、本研究では生成 AI を外部 API として利用する構成を採用した。これは、生成モデル自体をローカルに実装・訓練するのではなく、外部の高度な生成 AI をクラウド経由で呼び出すことで、学習者の端末に負荷をかけず、かつ最新の AI 機能を活用可能にするという利点を持つと考えられる。

さらに、API ベースの設計をすることで、教師モードやログ管理機能といった補助的な教育支援機能を柔軟に設計・追加できる構造的利点がある。教育的な制御性や発展性の観点から、外部 API の活用は本プロトタイプにおける合理的な選択であると考ええる。

3.2 Gemini API の採用理由と技術的特徴

本研究では、プロトタイプ構築に際して複数の生成 AI API を試用し、最終的に Google 社の Gemini API を採用した。その選定には、技術的な使いやすさに加え、教育現場での利用を想定した支援的環境の設計を念頭に置き、導入や継続利用の観点からの検討を行った。

初期段階では、OpenAI 社の提供する ChatGPT API を利用し、Colab 上での接続と動作確認を行った。OpenAI API は高精度な生成性能と多様なモデルの提供が魅力であり、教育利用においても十分な可能性を持っている。しかしながら、本研究の対象である中学生の段階的・安全な AI 活用環境の構築を目指す中で、以下のような課題が明らかになった。

第一に、無料プランの制限が非常に厳しく、OpenAI の無料利用枠は一日の使用回数やトークン数に上限が設けられており、数回の試行で制限に達することが多かった。とくに、API 経由での利用は ChatGPT のウェブ版とは異なり、無料枠が限定的かつ再取得が難しいため、中学生が繰り返し試行錯誤しながら学ぶような学習設計には適しにくいと判断した。

第二に、API キーの取得手続きやドキュメントの記述がやや技術者向けであり、プロンプト送信の際のフォーマットやモデル指定において混乱が生じやすかった。教育者や若年の学習者が扱うには、導入時のハードルがやや高い印象を受けた。これに対し、Google 社が提供する Gemini API は、Google Cloud Console での統合管理が可能で、Google アカウントを通じての認証や設定が行いやすいという利点があった。多くの学校で導入が進んでいる Google Workspace for Education との親和性や、運用面での柔軟性を評価した結果、教育現場への展開を視野に入れた際の実現可能性が高いと判断した。さらに、Gemini API ではテキスト生成に加え、画像入力など複数のモダリティに対応した生成 AI モデル（例：gemini-pro-vision）が利用

可能であり、将来的な教材開発や環境拡張にも柔軟に対応しやすい。また、筆者自身が OpenAI API と Gemini API の両方を用いて実装・動作検証を行った結果としても、Gemini の方が教育的文脈における接続性・操作性・継続性の面で優位であると判断した。

以上の理由により、本研究における生成 AI 活用の基盤には Gemini API を採用することとした。

研究初期には、OpenAI API の Chat Completion を用いた実装を試みたが、バージョン 1.0 以降で廃止された旧 API 構文を参照していたため動作せず、明確なエラーが表示された（例：You tried to access openai.ChatCompletion...）。また、Colab 環境においても、openai ライブラリのバージョン確認や固定インストール（例：!pip install openai==0.28）を行わなければならない、導入の難易度が高いと分かった。これらの経験は、教育現場で中学生が生成 AI を活用する際、技術的トラブルの発生リスクが高く、指導者の技術負担も大きいことを示唆している。

一方、Gemini API においても初期実装段階では複数の課題が発生した。たとえば、REST API に対して Bearer トークンではなく API キーを直接渡して認証エラー（401）が発生したり、models/gemini-pro が v1 beta で使用できないといった仕様面での混乱が見られた。また、Google Colab のキャッシュやセッション再起動のタイミングが動作に影響を及ぼすことも確認された。しかし、こうした課題はいずれもドキュメント確認や実行ログの精査を通じて解決が可能であり、最終的には安定した動作が確認できた。

これらの実装経験を通じて、Gemini API は教育利用を前提とした活用基盤として、柔軟性と安定性のバランスが取れており、学習者や教員の技術習熟度に応じて段階的に導入できるという点で、現実的かつ有効な選択肢であると判断した。

4. 初期実装と失敗の記録

本章では、生成 AI 支援システムの初期実装過程において実際に直面した技術的な課題と失敗事例を記録し、その内容を時系列に沿って整理する。特に、API ベースで構築することによって生じた実装上の混乱や仕様の読み違いは、教育現場に導入する際の再現性や安全性に関わる問題でもあるため、単なる技術トラブルとして片付けるのではなく、学習者や教員が遭遇しうる「つまずき」の予測とその支援設計への示唆として捉える。

以下では、OpenAI API を用いた初期実装の試行から始まり、最終的に Gemini API での安定動作に至るまでの失敗と修正の過程を記述する。

4.1 OpenAI API による初期実装の失敗と課題

初期段階では、ChatGPT を提供する OpenAI API を用いて、Google Colaboratory 上での生成 AI システム構築を試みた。しかし、いくつかの技術的な障壁に直面し、安定した動作に至らなかった。以下に、代表的な失敗例を挙げる。

(1) 旧 API 構文の使用によるエラー

Web 上の参考コードを元に `openai.ChatCompletion.create(...)` を実行したところ、OpenAI の API v 1.0 以降でこの構文が廃止されていたため、明確なエラーが返された。

エラー内容 : You tried to access openai.ChatCompletion...

このように、API の構文変更がドキュメント外で起こっていた場合、技術初心者や中学生がコードを模倣するだけでは動作しないケースが多く、教材としての信頼性や再現性に課題があると感じた。

(2) ライブラリバージョンの依存によるトラブル

Colab において openai ライブラリのバージョンが自動で適切に設定されず、旧仕様のコードがエラーと表示されることがある。この解決には、手動で下記のような操作が必要であった。

```
!pip show openai
!pip install openai==0.28
```

このように、環境構築の不透明さや細かい調整が必要になる点は、学習者への展開において障壁となりやすい。

(3) 無料枠の即時消費による継続不能

API キーを取得して動作確認を試みたが、数回のリクエスト実行で無料枠を使い切り、その後は一切の応答が得られなくなった。再取得は容易でなく、継続的な学習活動には不向きであると判断した。

以上、初期実装の段階から、OpenAI API による実装は教育現場での安定運用には向かないと結論づけ、別の選択肢として Gemini API の検討に移行した。

4.2 Gemini API 実装における技術的課題と対応

OpenAI API による実装が困難であることが判明した後、Google 社の Gemini API を用いた実装に移行した。Google Cloud Console による API キー管理や、Google Colaboratory との親和性の高さから、教育現場に導入するうえで有望と期待されたが、初期段階では複数の技術的な課題に直面した。以下では、それぞれの課題とその解決過程を記録する。

(1) 認証エラー : Bearer トークンの誤使用と API キーの形式不一致

初期実装では、Gemini API の REST エンドポイント (v1/models/gemini-pro:generateContent) に対して、API キーをヘッダに Authorization: Bearer <API キー> という形式で送信した。しかし、Google Gemini API では REST API における認証に OAuth 2 トークンが必要であり、通常の API キーではこの形式は使用できない。そのため、以下のようなエラーが発生した。

エラー内容 : 401 - UNAUTHENTICATED: Request had invalid authentication credentials.

この違いを理解していなかったことがエラーの原因であり、認証形式の明確な区別と設計判断の重要性が浮き彫りとなった。対応としては、REST API 経由の実装を中断し、公式にサポートされている

google.generativeai ライブラリを使用した Python 実装に切り替えることで解決を図った。

(2) モデル名の誤指定と API バージョンの不整合

API バージョン v 1 beta を指定した状態で gemini-pro モデルを呼び出したところ、指定したモデルが存在しないというエラーが発生した。

エラー内容 : models/gemini-pro is not found for API version v 1 beta

この問題は、指定したバージョンに対応する利用可能モデルを事前に確認していなかったことに起因していた。解決には <https://generativelanguage.googleapis.com/v1/models?key=...> を用いて、利用可能なモデル一覧を取得し、適切なモデル（例 : gemini-1.5-pro-002）を指定することによって正常動作が確認された。

(3) Google アカウントの制限によるアクセス不可

API キーを教育用の Google アカウント（Google Workspace for Education）で発行した場合、一部のモデルが使用できないことが判明した。この問題は、個人用アカウント（@gmail.com）に切り替えることで解消されたが、教育現場での利用を前提とする場合、Google アカウントの権限設計への配慮が必要であることが示唆された。

(4) Colab セッションのキャッシュによる不整合

Gemini 用ライブラリ（google-generativeai）を更新しても、Colab セッションに古いキャッシュが残っていたことで、意図したバージョンが反映されなかった。通常の pip install 後にランタイムを再起動せずにコードを実行したため、意図した変更が反映されなかった。

解決策 : os.kill(os.getpid(), 9) を実行し、強制的にセッションをリセットすることで、ライブラリのバージョン整合が取れた。

以上のように、Gemini API を用いた初期実装においても複数の技術的課題が発生したが、それぞれの問題に対しては実行ログや公式ドキュメントの精査を通じて原因を特定し、段階的に解決を図ることができた。表 1 に API 実装における主な失敗とその原因・対処を示す。教育現場での活用を前提とする場合、こうした「つまずきポイント」を予見的に記録・整理しておくことは、再現性ある教材設計や教員支援機能の実装において極めて有効である。

表 1 API 実装における主な失敗とその原因・対処

API 種別	失敗内容	主な原因	対処方法
OpenAI	旧 API 構文の使用	API バージョン 1.0 以降で構文が廃止されていた	最新 API 仕様に基づきコードを修正
OpenAI	ライブラリのバージョン依存	Colab 環境で旧バージョンが使用されていた	明示的にバージョンを固定して再インストール
OpenAI	無料枠の即時消費	トークン上限が非常に小さく、再取得が困難	別 API (Gemini 等) への移行を検討
Gemini	認証形式の誤使用	REST API に OAuth 2 が必要だが API キーを使用	Python 用ライブラリに切り替え
Gemini	モデル指定の不整合	v 1 beta では利用できないモデルを指定した	利用可能モデル一覧を API で事前取得して確認
Gemini	教育用アカウントで利用不可	教育用アカウントに一部制限があった	個人用 Google アカウントで API キーを取得
Gemini	Colab のセッションキャッシュ	ライブラリ更新後も古いバージョンが残っていた	セッションを強制的に再起動

5. 技術的改善の過程と考察

5.1 技術的課題の発見と修正プロセス

第 4 章で述べた通り、本研究では複数の実装上の問題に直面したが、それぞれの課題に対して段階的な調査と修正を行うことで、最終的に安定した動作を得ることができた。以下では、どのように問題を発見し、修正に至ったのか、そのプロセスを整理する。

まず、各種エラーの発見には、Google Colaboratory 上のログ出力と、エラーメッセージの読解が重要な手がかりとなった。たとえば、認証エラー (401 UNAUTHENTICATED) では、エラーメッセージの中に「Expected OAuth 2 access token」という記述があり、API キーではなく OAuth 2 トークンが求められていることを読み取ることができた。また、モデル指定のエラーでは、「models/gemini-pro is not found for API version v 1 beta」というエラーメッセージを受けて、API バージョンと利用可能モデルの対応関係を調査する必要があると判断し、Gemini の公式ドキュメントおよび API から取得可能なモデル一覧を確認する作業を行った。

さらに、Colab 環境でのトラブルでは、ライブラリのキャッシュが影響している可能性を疑い、明示的に `!pip install` を再実行した上で `os.kill(os.getpid(), 9)` によってセッションを強制的に再起動する方法を試すことで、動作不全を解消することができた。これらの過程はすべて、システム開発の技術的対応であると同時に、教育支援システムの設計においても重要な示唆を与える。すなわち、学習者が AI を活用する環境において、何らかのトラブルやエラーが生じた際に、それをどのように発見し、理解し、修正できるかというプロセスそのものが、情報活用能力の育成につながる可能性を持っている。

5.2 教育現場への応用可能性

第4.1節で述べたような実装トラブルの発見と修正のプロセスは、単なる開発作業ではなく、教育的な視点からも極めて価値の高い活動である。特に、本研究で対象とする中学生段階の学習者にとっては、「エラーから学ぶ」という経験は情報活用能力や問題解決力の育成に直結する。

例えば、実装中に表示されるエラーメッセージを読み解き、その意味を調べ、原因を推測し、ドキュメントを参照して修正案を考えるという一連の流れは、まさに ICT リテラシーや探究的学習の実践そのものである。このようなプロセスは、生成 AI に対して受動的に使うだけでなく、能動的に使いこなす姿勢を育むことに資する。また、指導者にとっても、生徒がどのようなトラブルに遭遇しやすいかを事前に理解しておくことで、適切な支援や指導が行いやすくなる。たとえば、プロンプト送信がうまくいかないときに「認証形式を確認しよう」「モデル名が正しいか調べよう」といった観点からヒントを与えることで、生徒自身の自立的な学習を促すことが可能になる。

さらに、本研究で得られた失敗事例や対処法を活用すれば、学習者向けの「トラブル対応の整理表」や「原因予測を通じた思考支援教材」といった形で、学びに転換する教材開発も検討できる。こうした教材は、生成 AI を扱う授業そのものを、論理的思考力・読解力・情報調査力といった複合的な力を育成する場となり得る。

5.3 指導者のための支援設計

これまで述べた失敗事例と改善過程は、教師が教育支援システムを運用する際の設計的示唆も多く含んでいる。たとえば、指導者自身がエラーや設定ミスに即座に気づけるような、操作状況の可視化や確認を支援するインタフェース設計が求められる。エラー表示の日本語化や簡易モードでの実行結果のハイライト表示、推奨プロンプトの例示などは、初学者を支援する指導者にとって有用な工夫といえる。

また、学習者の生成 AI の活用状況を指導者が把握できるようなプロンプトや出力の利用傾向を俯瞰的に確認できる仕組みや、出力内容の中に含まれる不適切な語句を検出し、指導支援に活かすような利用モニタリング支援の構想も検討に値する。これにより、指導者は生徒の生成 AI 利用状況を包括的に把握でき、適切なフィードバックや介入を行うことができる。さらに、実装時の問題を事前に指導者側で把握しておくことで、授業中のトラブル対応が迅速に行える。たとえば、「モデル名の選択ミス」や「API キーの貼付忘れ」など、頻出するエラーを事前に共有するガイドラインやチェックリストの整備も、教員研修やマニュアル化の一部として有効である。

以上のように、本研究で得られた知見は、学習者だけでなくその支援者である指導者にとっても現場実装に向けた重要な設計的インサイトを提供するものである。

6. まとめと今後の課題

本研究では、中学生が安全かつ教育的に生成 AI を活用することを目指したプロトタイプ開発において、初期の実装段階で発生した数々の技術的課題と、それに対する具体的な対応過程を記録・考察した。とりわけ、API 構文や認証形式の誤解、モデル名の指定ミスなど、初学者や教育現場で起こりうる「つまずき」を明確化し、それらの技術的改善を通じて教育支援設計への示唆を得たことは、本研究の大きな成果である。なお、教育用 Google アカウントによるアクセス制限に関しては、学校独自で発行するアカウントと大学

の研究機関を通じて発行するアカウントの双方を用いて検証を行ったが、モデル使用の可否や制限内容に差異が見られた。これについては、十分なサンプル数と設定条件をもとにしたさらなる検証が必要である。

今後は、本研究で構築したプロトタイプを基盤として、利用履歴や対話記録に基づいた安全性の確認・支援の仕組みや、学習者の状況に応じたプロンプト活用支援や運用補助を可能とする柔軟な支援環境の整備を検討していく予定である。これにより、教員と学習者の双方が安心して生成 AI を活用できる「教育現場に最適化された生成 AI 支援環境」の実現を目指す。さらに、実証実験やユーザビリティ評価、教育的効果の検証へと発展させていく予定である。

参考・引用文献

- 1) 松田孝・景井美帆・亀井俊之・桑村海光・人見久城・磯部征尊・大森康正・山崎貞登:
STEM/STEAM 教育からの小学校段階における AI リテラシー育成のための教材開発と実践,
上越教育大学研究紀要, vol.40 (2), pp.27-38, 2021.
- 2) ・文部科学省 (2024) 「初等中等教育段階における生成 AI の利活用に関するガイドライン Ver. 2.0」,
https://www.mext.go.jp/content/20241226-mxt_shuukyo_02-000030823_001.pdf, 2024 年 4 月 3 日取得.
・文部科学省. (2023). 初等中等教育段階における生成 AI の適切な活用に関する暫定的なガイドライン (参考資料 3) [令和 5 年 7 月 18 日 義務教育の在り方ワーキンググループ (第 7 回)],
https://www.mext.go.jp/content/20230718-mtx_syoto_02-000031167_011.pdf, 2024 年 4 月 3 日取得.
- 3) 厚生労働省 (2023) : 「資料 1 生成 AI の技術動向と影響」, <https://www.mhlw.go.jp/content/11601000/001125241.pdf> <https://kncoder.net/>, 2024 年 4 月 3 日取得.
- 4) Google (2024) 「Gemini API documentation」 Google AI, 最終アクセス日時 (2025 年 4 月 20 日),
<https://ai.google.dev/gemini-api/docs>
- 5) OpenAI (2023) 「Best practices for prompt engineering with GPT」 OpenAI Documentation, 最終アクセス日時 (2025 年 4 月 20 日), <https://platform.openai.com/docs/guides/gpt-best-practices>

Challenges and Design Considerations in the Initial Implementation of a Generative AI Educational Support System for Elementary and Junior High School Students

Abstract

In this study, we designed and implemented a generative AI support system to help elementary and junior high school students use AI safely and meaningfully in educational contexts. While generative AI has significant potential to support students' expression, thinking, and inquiry, it also poses notable risks, such as overreliance, misinformation, and difficulty with ethical judgment-particularly for younger learners with developing digital literacy.

Our prototype was built using Google Colaboratory and the Gemini API, with careful attention to balancing technical feasibility and educational applicability. During the implementation process, we encountered various technical issues, including authentication errors, model misconfiguration, and environment-related conflicts. We analyzed and interpreted these challenges not only as technical failures but also as sources of insight for designing effective educational applications.

We further examined how these issues could inform instructional strategies for teachers, such as anticipating common student mistakes and supporting autonomous learning. Based on this experience, we suggest that documenting and reflecting on implementation failures contributes to the creation of more robust, reproducible, and pedagogically meaningful generative AI learning environments.

This work forms part of a broader research project that aims to design AI systems for young learners that are sustainable, expandable, and educationally sound.

Key Words : Generative AI, Educational technology, Elementary and junior high school education, Technical challenges.