PrepGPT: A Localized, Multi-Agent AI Tutor Fine-Tuned with Dynamic Entropy GRPO for Empathetic and Diverse Explanations

Himaloy Mondal, Kethu Charan Kumar Reddy & Advick Shukla

Received September 5, 2025
Accepted December 5, 2025
Electronic access December 15, 2025

Academic stress and inefficient study habits are major challenges for students worldwide, yet personalized tutoring remains inaccessible for many learners. Current AI tutors mainly rely on imitation learning, which allows them to reproduce expert explanations but limits their ability to adapt to diverse student needs. In response, this study introduces PrepGPT, a locally deployed, privacy-first AI-tutor focused on student-sensitive and curriculum-aligned support aimed at enhancing adaptive learning. Our primary contribution is the development of the Emergent Machine Pedagogy (EMP) conceptual framework, which is a theory of how machine learning can be adapted to recognize and respond to individual student learning patterns; we refer to it as Dynamic Entropy Group Relative Policy Optimization (DE-GRPO): an optimization method that personalizes educational feedback by learning from diverse student interactions. To ensure practical relevance, we conducted two human surveys: one involving 70 students and another involving 20 practicing educators who performed blind comparative assessments against baseline models. In small-scale experiments, DE-GRPO produced explanations that were more varied and sometimes more effective than standard methods. We integrate diversity-aware training of the Teacher via DE-GRPO and provide runtime logs demonstrating reward progression and entropy control. We further validated the framework by developing PrepGPT, which demonstrates key features such as adaptive explanations, multiple solution paths, and handwritten study note generation in the student's own handwriting; all while operating on consumer hardware without cloud dependence. These results provide early, framework evidence that AI tutors may be able to move beyond simple imitation and move towards more adaptive strategies.

Keywords: Emergent Machine Pedagogy (EMP); Dynamic Entropy Group Relative Policy Optimization (DE-GRPO); Curriculum-Aligned Support; Adaptive Explanations; Handwritten Notes Generation; Imitation Learning; Reinforcement Learning; AI in Education.

Introduction

Academic stress and inefficient study habits are prevalent issues affecting students from all academic backgrounds[1]. The increasing demands of coursework, the overwhelming volume of study materials, and peer pressure contribute to challenges in academic performance and well-being, even raising college dropout rates[2,3]. A 2023 study of 320 undergraduate nursing students in Mangalore found that 65.6% experienced moderate stress and 31.9% experienced high stress, with higher stress levels directly associated with poorer grades[4]. Personalized support such as one-on-one tutoring is linked to higher achievement, yet limited accessibility creates inequities. Thus, there is a pressing need to create accessible solutions that provide personalized study support at scale[5]. A semester-long study was conducted at UniDistance Suisse, an AI tutor app provided to psychology students taking a neuroscience course. It generated microlearning questions from pre existing materials and the tutor maintained a dynamic neural network of every student's grasp of concepts. This shows implementation of dynamic retrieval practice and personalisation.

Artificial intelligence tools like ChatGPT and Bard are increasingly being adopted and used in education, showing strong capability in domains such as grammar correction, structured factual recall, and real-time Q&A for well-defined queries. In these contexts, LLMs demonstrate a high degree of reliability—especially when their outputs can be verified against authoritative sources, or when retrieval-augmented techniques are used. However persistent issues remain, ChatGPT-3.5 fabricated 55% of references and introduced substantive errors in 43% of non-fabricated citations[6]. Similarly, a systematic review found hallucinated references in 39.6% of ChatGPT-3.5 outputs and 91.4% of Bard's[7].

Such unreliability makes them unsuitable for education where factual precision is essential. While platforms like Socratic, Ada, and Habitica support learning, no peer-reviewed evidence shows they integrate curriculum analysis, generate adaptive study aids such as flashcards, or incorporate essential privacy and transparency mechanisms[8,9]. Thus the role of LLMs in the education sector is key, but comes with many limitations too[10].

Large LLMs also struggle to emulate flexible, relational reasoning. Although they achieve human-level performance on some analogy tasks, researchers debate whether this reflects genuine understanding or surface-level pattern matching[11]. Chinese chatbot DeepSeek, though cost-effective, faced restrictions or bans in several countries due to privacy and security concerns[12] and many other privacy risks are included[13]. Even the most advanced LLMs remain limited in explaining complex concepts in diverse ways that adapt to individual learning needs[14]. Their reasoning is primarily associative rather than relational, limiting their ability to reframe misunderstood concepts or synthesize new explanatory analogies[15]. Moreover, most AI applications in education focus on surface-level personalization rather than aligning content with specific academic goals[16].

Despite the growth of commercial AI tools, little research explores open-source, locally deployed systems that are privacy-conscious and curriculum-specific. This gap—the lack of open-source, curriculum-aware, and privacy-preserving tutoring agents—is what drives our work. With PrepGPT, we hope to reduce exam-related stress and improve study efficiency by creating an AI-powered chatbot that works locally on the user's device, ensuring data both privacy and security. Designed with ethical principles in mind, PrepGPT gives students full control while analyzing their course syllabi to highlight key concepts and create personalized, adaptive flashcards to make learning more effective. To address LLMs' lack of pedagogical flexibility, we introduce a Teacher LLM fine-tuning module designed to synthesize explanations, analogies, and counter-examples tailored to student needs. Unlike other template-based outputs, the Teacher LLM anticipates misconceptions and reframes explanations, simulating human-like instructional nuance for the user.

Built using a modular LangGraph architecture, the chatbot leverages open-source LLaMA models for local inference, ensuring both privacy and accessibility. We also introduce a framework simulating two interacting agentic systems designed to co-evolve through instructional dynamics. In our setup, one model generates challenging questions while the Teacher LLM is rewarded for resolving confusion.

Trained pedagogically, the Teacher LLM learns to explain concepts clearly and anticipate whether responses effectively address misunderstandings or not. Interestingly, the AI sometimes appeared able to anticipate the effectiveness of its explanations. Supporting this framework is a self-synthesizing automation pipeline which manages training, question generation, feedback, and refinement—attempting to create a framework for scalable, curriculum-aware tutoring agents.

By uniting explainability, adaptivity, and ethical design, this study aims to make organized, high-impact study tools accessible to students lacking formal tutoring, promoting educational equality. The modern AI era has been defined by imitation learning, where models replicate human-generated patterns. This has produced remarkable capabilities, but it also reveals a fundamental limitation: an imitation system is a high-fidelity mirror—it cannot create beyond its data. We define this as the **Imitation Efficacy Ceiling (IEC)**; imitation alone cannot systematically exceed the best available expert demonstrations, explained briefly and formally further explained in the theoretical framework subsection of the methods section.

This paper explores a framework that we call **Emergent Machine Pedagogy (EMP)**. We suggest co-evolving agents could discover new teaching strategies through guided interaction, much as self-play enabled AlphaGo to discover superhuman strategies in defined games. A principled "self-teaching game" may unlock autonomous discovery of novel principles in pedagogy, reasoning, and design. This reframes learning from imitation toward goal-oriented discovery.

The core limitation of current AI tutors remains their reliance on imitation learning, constrained by the IEC: combining existing expert examples cannot yield performance beyond the single best expert. To enable inventive AI, systems must venture beyond this ceiling.

This paper introduces a complete framework to achieve this. First, we present the conceptual framework of Emergent Machine Pedagogy. Second, we introduce our novel algorithm, Dynamic Entropy Group Relative Policy Optimization (DE-GRPO), translating EMP principles into reinforcement learning practice. Finally, we share preliminary results in a controlled environment suggesting DE-GRPO may outperform imitation-based methods.

Methods

An experimental design was conducted to evaluate the performance of the PrepGPT AI tutoring system. Our approach is 2 fold; Introduce a new theoretical framework Emergent Machine Pedagogy (EMP) which motivates the core algorithm of our system, secondly detail a full implementation of the PrepGPT application, a localized, privacy first, multi agent model designed to bridge the educational divide for students from under resourced backgrounds. This section architects the system architecture, the core theoretical algorithm, key features like handwriting notes generation and ethical inclusions guided this research.

Our framework is built upon the COGNITA stochastic game formalism, a multi-agent environment designed to model pedagogical interactions. This framework provides 5 principles that: Defines the limits of the old paradigm, proving the potential of the new, describing the mechanism of discovery, guaranteeing its stability and alignment and proving its components are essential.

Theoretical Framework: The Quintet for Emergent Pedagogy

To model the complex, interactive nature of teaching, a simple Markov Decision Process (MDP) is insufficient, as it typically frames the problem from a single agent's perspective. We instead adopt a multi-agent stochastic game formalism, COGNITA[17] (Mathematical specification of COGNITA: See Appendix A.2 for the concrete instantiation used in our PedagogyEnv). This choice is deliberate: it allows us to explicitly model the distinct, and sometimes competing, objectives of the Teacher (pedagogical efficacy), the Student (knowledge acquisition), and the Verifier (objective grounding). This multi-agent view is critical for capturing the dynamics of guided discovery, where the Teacher's policy must adapt to a changing Student state, a problem that cannot be fully captured by a static environment.

Our DE-GRPO algorithm was designed to reflect ideas from our conceptual framework, Emergent Machine Pedagogy (EMP)[18,19]. This framework is built on a quintet of guiding principles that define a logical arc from the problem to a potential solution. We outline the reasoning behind these principles and connect them to established ideas in reinforcement learning and pedagogy[20].

Impossibility: The IEC. This principle establishes that an AI learning only from expert examples is forever capped by the performance of the best single expert[21,22]. It defines the problem we must solve. Illustrative example: A warehouse robot is trained with behavioral cloning from expert teleoperation videos to pick items from bins and place them on a conveyor, initially matching expert moves on common parts and lighting but never interacting with the environment during training[23]. As more of the same videos are added, success rises quickly on standard cases but then plateaus at a high-yet-suboptimal rate, because rare failure modes (crumpled packaging, occlusions, slippery surfaces) are underrepresented and the policy encounters unseen states where compounding errors amplify, forming the imitation ceiling[22,24]. The plateau persists because demonstrations omit crucial latent information (tactile cues, precise friction, fine-grained depth), so copying observable trajectories cannot resolve ambiguities or recover when off-distribution. While such systems can equal expert humans, they cannot systematically surpass them or discover strategies outside human experience. To create inventive AI systems capable of novel problem-solving, we must move beyond imitation[25].

Possibility: The Discovery-Efficacy Tradeoff. For breaking the ceiling, an agent must explore novel strategies. This exploration always carries the risk of temporary failure, but it is the necessary price for discovering a superior strategy for the agent. It proves a solution is possible, but costly. Innovation is possible, but since experiments can fail, we must control and encourage productive exploration.

Mechanism: The Critical Diversity Threshold. Discovery is not gradual; it is a phase transition. Below a critical level of strategic diversity, an agent remains trapped. Above it, breakthroughs become possible[26,27]. This tells us how to enable discovery, by actively managing diversity. This highlights a practical mechanism: explicitly manage diversity so the system reaches the regime where breakthroughs occur[28].

Robustness: The PAC-Verifier Guarantee. Invention without grounding is hallucination[29]. This principle guarantees that if the agent's discoveries are judged by a reliable "Verifier," the resulting strategies will be genuinely effective, not just creative fictions. This ensures our agent learns useful things. Even with rich discovery, inventions must be grounded — hence the Verifier is necessary to ensure practical value.

Necessity: The No Free Lunch principle for Pedagogy. This final principle proves that the two core components, a mechanism for novel strategy generation (Creativity) and a grounded Verifier (Reality Check), are both essential. Without both, an agent is probably doomed to remain an imitator. Taken together, these results imply that any successful pedagogical agent must jointly implement both creativity and a reliable reality check.

Data Acquisition: Data Cleaning, Annotation, and Preprocessing Strategies

To enable intelligent, pedagogically aligned responses, our system integrates three distinct knowledge pipelines that together support foundational learning, up-to-date reasoning, and fine-tuned teaching behaviour. These pipelines contribute to different layers of the chatbot's understanding and adaptability.

Curated Academic Knowledge Base (Textbook Embedding): This is the system's primary source of domain-specific, structured information. The pipeline begins with the upload of the course textbooks or reference materials in PDF format. A custom Python script is used to extract clean textual data from the PDFs. At present, advanced OCR (Optical character recognition) based techniques are not employed, but this is adequate for editable digital files, but less robust for scanned or image-based documents. In future versions, an OCR module will be incorporated to ensure accurate extraction from scans, photos, and non-selectable PDFs—such as handwritten notes or digitized textbooks—thus closing this critical gap for comprehensive, curriculum-aligned learning. Extracted text is then segmented into semantically coherent units, typically paragraphs or sentences, using basic English syntax rules to ensure that related concepts remain grouped; Each chunk is converted into a high-dimensional vector embedding and stored in a ChromaDB instance. This is the framework for enabling fast, context-aware retrieval during the question-

Table 1 EMP principle to a plain-language meaning, and a short classroom vignette.

Principle	One-line meaning	Real-life teaching example
Impossibility (IEC)	Learning only from expert examples cannot exceed the best expert.	A novice teacher who only copies a master teacher never develops a better method.
Discovery–Efficacy Tradeoff	To surpass experts you must explore new strategies, accepting short-term failures.	A teacher experiments with a flipped-classroom activity; student scores dip initially but later improve for deeper understanding.
Critical Diversity Threshold	Breakthroughs occur only after enough diversity of strategies is present — a phase transition.	A department encourages many different lesson formats; only after several novel trials does one format produce large, sustained gains.
PAC-Verifier Guarantee	Creative strategies must be objectively validated, or they risk being hallucinations.	A teacher's new technique is validated through blind assessments and control groups before being adopted school-wide.
No Free Lunch for Pedagogy	Both creativity (novel strategy generation) and grounding (objective verification) are essential.	A school requires teachers to both propose new lesson designs and run pilot evaluations before scaling them.

answering process.

Dynamic Web Content Retrieval (External Knowledge Augmentation): To solve the factual cutoff and limitations of pre-trained LLMs, our system dynamically incorporates up-to-date web knowledge. The process begins with a user query triggering the Google Search API, returning relevant links in real time, top-ranked URLs are processed using Crawl 4AI, which downloads and cleans webpage content by removing vague and biased elements. The cleaned text is then segmented, embedded, and stored in a session-specific FAISS in-memory vector store. This framework enables rapid retrieval of relevant information without relying on persistent storage, therefore minimizing redundant API calls and causing enhanced user privacy.

Instructional Fine-Tuning Pipeline (Simulated Teaching Behavior): Beyond retrieving factual knowledge, our system also aims to simulate "how good instruction is delivered?". This is achieved through a novel framework approach for teacher-informed fine-tuning. Due to the impracticality of collecting thousands of real teacher–student interactions, we used large language models (LLMs) to scale our dataset. These LLMs were prompted with: Core textbook content, and the models generated synthetic instructional examples across multiple subjects and learning styles. While synthetic data has limitations, it allows us to fine-tune smaller, privacy-preserving models for task-specific educational performance. This approach aims to approximate the output quality of larger models while avoiding the high data, compute, and ethical costs typical of large-scale AI training.

Scaling properties for more capable LLMs

To ensure the long term relevance of our framework in an era of rapidly scaling models, we also provide formal guarantees about its behaviour with more capable agents. We also suggest, in theory, that if an AI tutor is more capable, it should at least match or improve upon the performance of a less capable tutor. This provides a theoretical foundation to the scalability of our approach, ensuring that improvements in model architecture will translate to a higher potential for inventive discovery within our framework.

Computational Framework and Models

The core experiments in `run_full_suite.py` compare three main algorithmic instantiations. Before presenting our solution, it is crucial to establish why standard algorithms are ill-equipped for the inventive pedagogical task we have defined. **Proximal Policy Optimization (PPO):** While a powerful reinforcement learning algorithm, PPO's exploration is typically driven by entropy over the action space, which encourages random token sequences[30]. In a natural language domain, this often leads to incoherent or grammatically incorrect 'exploration' rather than semantically meaningful new teaching strategies. It effectively explores syntax, not concepts, making it unsuited for discovering novel pedagogical analogies. **Naive Evolutionary Search (GRPO-Normal):** Our baseline evolutionary method, which selects purely for reward (efficacy), is highly susceptible to premature convergence. In our experiments, it quickly finds a locally optimal 'good enough' strategy and collapses the population's diversity, thereby eliminating the very mechanism, strategic variety, needed to make inventive leaps. These fundamental limitations demand an algorithm that explores not randomly, but strategically, and that treats diversity not as a byproduct to be maximized, but as a critical resource to be actively managed. This directly motivates the design of DE-GRPO.

Algorithms and Metrics

The Supervised Fine-Tuning (SFT) baseline represents the best static policy discoverable by imitating the expert dataset. It functions by selecting the expert examples that are most similar to a target concept, thereby establishing the practical Imitation Ceiling for our experiments.

The Generative Reward Policy Optimization (GRPO) baseline is a naive evolutionary search agent. It explores by generating candidate responses and selecting purely for the highest reward (efficacy), without any explicit diversity management.

DE-GRPO, the algorithmic incarnation of our framework, is aptly termed *The Principled Inventor*. It executes a state-aware scoring function that embodies the principles and constructs an actionable, measurable objective for the agent, balancing high performance and imaginative exploration. The final score assigned to a candidate response is:

$$\text{Score} = R(\pi) + \alpha \cdot D(\pi) \tag{1}$$

Formally:

$$R_t^T = r_t^{\text{eff}} + \lambda_t d_t \quad \text{with} \quad \lambda_t = \alpha(1 - r_t) \tag{2}$$

This scoring is a direct execution of our framework, in which $R(\pi)$ is the efficacy reward provided by the Verifier agent. This directly satisfies the PAC-Verifier Guarantee, ensuring that the agent's creative exploration is grounded in what is effective and avoids ungrounded hallucination. $\alpha \cdot D(\pi)$ is the term corresponding to the dynamic diversity bonus, intended to address the Discovery-Efficacy Tradeoff. The adaptive coefficient α increases dynamically when performance stagnates, adding novelty to help the agent cross the Critical Diversity Threshold and escape local minima where imitation-based methods would be trapped.

The following specification of the dynamic diversity coefficient α governs this adaptive behavior:

$$\alpha = \alpha_{\text{base}} \cdot (1 - \bar{R}(\pi)) \tag{3}$$

where $R(\pi)$ is the reward.

The theoretical definitions of efficacy and novelty are operationalised in our analytical script, `analyze_results.py`, in the following way:

Efficacy (Reward): This metric measures the quality of the generated explanation. We use the underlying LLM as well as a "student proxy" and test them with a short, standardized test derived from the explanation. Efficacy is the percentage of correctly answered questions. It constitutes a functional, objective metric of pedagogical efficacy. We also adopt entropy-regularized policy gradients to preserve calibrated stochasticity and stabilize updates during training[31].

Novelty: This metric is used to measure the originality of an explanation. It is computed as one minus the highest cosine similarity between the explanation's embedding and the embeddings of all examples in the expert dataset. A high novelty score indicates a generated strategy that is semantically at a marked distance from any given human data.

Model fine-tuning: Reinforcement Learning based approach

The team confirms that PrepGPT's Teacher LLM uses diversity-aware training implemented via Diversity-Enhanced Group Relative Policy Optimization (DE-GRPO) with dynamic entropy management to prevent policy collapse and encourage diverse, adaptive instructional responses. Figures 12 and 13 (Mentioned In Appendix: section B.4) show log snapshots from a complete DE-GRPO run, including iteration-level reward progression and diversity scores, with the final pass reporting Avg Reward = 0.6245 and Diversity = 0.1836, and the saved artifact 'results/entropy_results_1752552191.json' for reproducibility. Across iterations, multiple candidate responses are scored by an efficacy reward and a diversity term, and higher-scoring trajectories replace lower-scoring exemplars according to the DE-GRPO update, matching the selection rule described in this section. The DE-GRPO training run shown in Figs. 12–13 completed 10/10 iterations ('DE-GRPO Iterations: 100%') and archived outputs to `results/entropy_results_1752552191.json` for inspection.

System Overview and Architecture

In order to demonstrate the practical implementation of our proposed theoretical framework, we developed PrepGPT, a fully integrated AI tutor that can be deployed locally. The user interface is developed in React.js, while the request processing is handled by a Python Django backend. LangGraph is used to execute the main AI logic, as it allows the orchestration of a pipeline of modular agents for sequence tasks such as query classification, web searching, retrieval-augmented generation, and answer validation. In keeping with our user-centered design principles and as a step towards the vision of crafting educational tools that users can personally rely on, a novel module was developed that can generate study notes in the user's own handwriting. Such a design extends the concept of personal teaching aids. This model is a tangible proof-of-concept for our research and is aligned with privacy-by-design principles, as it operates fully on-device.

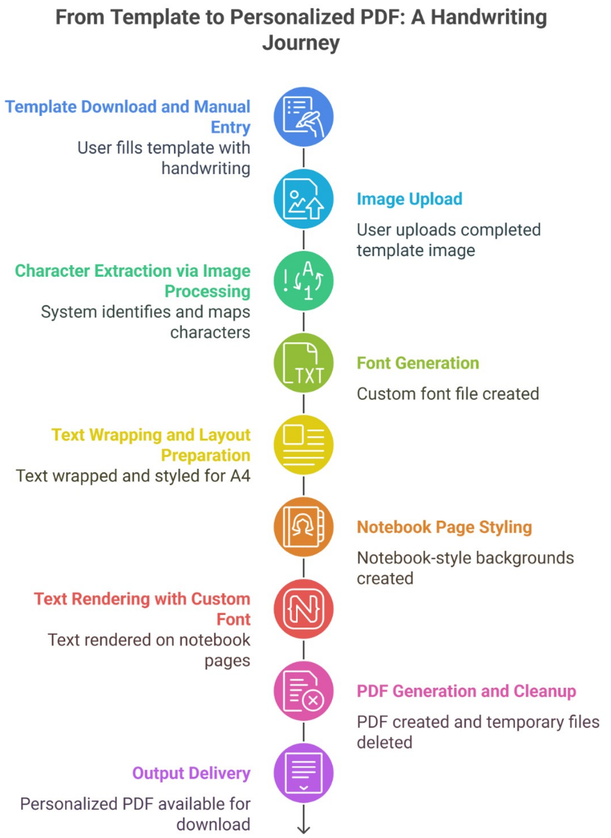
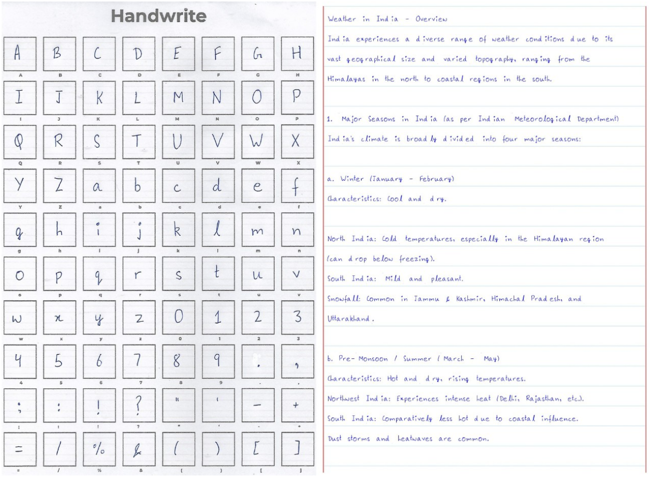

From Template to Personalized PDF: A Handwriting Journey
Template Download and Manual Entry
User fills template with handwriting
Image Upload
User uploads completed template image
Character Extraction via Image Processing
System identifies and maps characters
Font Generation
Custom font file created
Text Wrapping and Layout Preparation
Text wrapped and styled for A4
Notebook Page Styling
Notebook-style backgrounds created
Text Rendering with Custom Font
Text rendered on notebook pages
PDF Generation and Cleanup
PDF created and temporary files deleted
Output Delivery
Personalized PDF available for download

Fig. 1 Step by step journey from handwritten template to a downloadable personalized PDF, powered by image processing and custom font rendering.

Web Scraper with Crawl4AI

To automate the extraction of educational or domain-specific textual content from websites, so that we can use it to feed our model, we developed a customized web scraping pipeline using the Crawl 4I framework. While traditional scraping which is 'bs4' and 'requests' provide control over HTML parsing, they often require a lot of extensive manual intervention, especially while handling varying website structures, dynamic elements and deep internal linking. Moreover, 'requests' is not ideal for hierarchical or nested HTML parsing due to its lack of structural awareness, making it more error-prone for real-world documents. Crawl4AI eliminates these limitations, by using crawler recursively which explores internal links with intelligent prioritization based on content density, topic relevance, and HTML tag heuristics. It also normalizes the extracted text into well-structured paragraphs and sections, streamlining integration with downstream NLP modules like summarization, Q&A generation, and document embedding.



A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	a	b	c	d	e	f
g	h	i	j	k	l	m	n
o	p	q	r	s	t	u	v
w	x	y	z	0	1	2	3
4	5	6	7	8	9	.	,
;	:	!	?	"	'	-	+
=	/	%	&	(	)	[	]

Weather in India - Overview

India experiences a diverse range of weather conditions due to its vast geographical size and varied topography, ranging from the Himalayas in the north to coastal regions in the south.

1. Major Seasons in India (as per Indian Meteorological Department)
India's climate is broadly divided into four major seasons:

a. Winter (January - February)
Characteristics: Cool and dry.

North India: Cold temperatures, especially in the Himalayan region (can drop below freezing).
South India: Mild and pleasant.
Snowfall: Common in Jammu & Kashmir, Himachal Pradesh, and Uttarakhand.

b. Pre-Monsoon / Summer (March - May)
Characteristics: Hot and dry, rising temperatures.
Northwest India: Experiences intense heat (Delhi, Rajasthan, etc.).
South India: Comparatively less hot due to coastal influence.
Dust storms and heatwaves are common.

Fig. 2 Left: A completed handwriting sheet showing handwritten examples of uppercase and lowercase letters, numbers and symbols; Right: Handwritten notes in user's handwriting

To ensure that only reliable and educationally valuable content is used to train or fine-tune our models, we integrated a trust-based filtration mechanism into the Crawl4AI web scraping pipeline. This filtering step is designed to prevent the inclusion of pages that are misleading, overly subjective, or lacking in educational depth—issues that are especially common in open web crawling. The filter works in three main stages. First, it checks the source of each webpage to determine if it belongs to a trusted domain. This includes matching both general educational suffixes such as ".edu", ".ac.", ".edu.", ".edu.au", ".edu.sg", ".ac.uk", ".ac.in", ".ac.jp", ".ac.nz", ".ac.za", ".edu.in", ".edu.mx", ".k12.", ".sch.", ".gov", ".nic.", ".academy", ".school", ".education", ".college", ".institute", as well as a curated list of trusted education platforms and institutions. The list includes globally known websites like Khan Academy, Wikipedia, OpenStax, MIT OCW, NPTEL, and Coursera, along with regionally significant platforms such as Byjus, Doubtnut, Physics Wallah, Allen, NCERT, CBSE, and many others.

Second, the system performs a light-weight scan of the page's content for language patterns that are typically associated with sensationalism or personal opinion—phrases like "shocking", "unbelievable", "miracle", "exposed", "secret", "conspiracy", "scandal", "I think", "we believe", "in my opinion"—using regular expression checks. Third, it verifies that the page contains a sufficient amount of meaningful content by checking for a minimum character length, which helps avoid empty, stub-like, or purely navigational pages.

These signals are then combined into a single trust score (ranging from 0 to 1), where higher values indicate more credible and content-rich pages. Only pages scoring above a configurable threshold (default is 0.60) are allowed to continue into downstream processing tasks like summarization, Q&A generation, or document embedding. This mechanism is built to be easy to maintain: it provides a drop-in function (`basic_trust_check`) that integrates seamlessly with the rest of the scraping pipeline. Overall, this filtration process allows us to automatically collect large-scale educational content while maintaining control over the trustworthiness and pedagogical value of the material.

Protecting Web Interactions

In terms of security, special emphasis was placed on Cross-Site Request Forgery (CSRF) protection. Since our application accepts form submissions and file uploads from users, we needed to prevent unauthorized or malicious requests. Django provides built-in CSRF protection, which we configured to work with our React frontend by exposing the CSRF token via cookies and reading it in JavaScript before making requests. This token is then also sent along with every POST request and validated on the server. If a request lacks a valid CSRF token or originates from an untrusted source, Django automatically rejects it. This ensures that all data submissions, from form text to uploaded files, are coming from legitimate, authenticated users interacting directly with our frontend. Without CSRF protection, attackers could trick users into unknowingly submitting data or files through malicious scripts.

Ethical Considerations

PrepGPT is developed with robust ethical principles in mind, inspired by PRISMA's emphasis on transparency, UNESCO principles for ethical use in AI and also the unified framework of five principles in AI society[32]. Its main mission is keeping students safe with inclusive and accurate study materials while meeting international standards for protecting information with privacy-by-design and privacy-by-default approaches[33], thus trying to neutralize privacy threats posed by LLM's and cloud based deployment models[34]. By computing locally, the model limits data exposure while ensuring adherence to laws like GDPR, which imposes data minimization, consent, and erasure rights, and COPPA, which mandates parent consent for children under 13. Digital well-being is also a consideration for PrepGPT in recommending study breaks, reducing screen use, and promoting emotional health so it is an emotionally intelligent model for fairness and integrity in learning. Above all, it is designed to support, not replace, human interpersonal relationships in the learning process.

Evaluation of Methodology

Two-Part Validation Strategy

To test our ideas in a way that's both scientific and practical, we used a two-part strategy. This let us show two things at once: (1) how our algorithm actually works under controlled conditions, and (2) that it could be built into something useful in the real world.

The Controlled Python Environment: This was the "science lab" of our project. We built a simple but careful setup in Python that included the three main agents from our COGNITA framework: a Teacher, a Student, and a Verifier. In this space, we could run clean, head-to-head comparisons of teaching algorithms and measure two things very precisely: how effective the teaching was (Efficacy) and how different the explanations were from the usual expert dataset (Diversity).

The Real-World Prototype (PrepGPT): Of course, we didn't want this to stay stuck in theory. So we have built a working prototype called PrepGPT, which runs on normal computers. This tool shows that our ideas can actually be applied in practice, not just on paper. PrepGPT respects privacy from the ground up, is designed for fairness and inclusivity, and acts as a real example of how our algorithm could help students.

Experiment Setup

All of our experiments were run in the Python environment, focusing on a single teaching task: coming up with a new and effective explanation for entropy. We tested three main agents against our baseline:

- **SFT (The Imitator):** A baseline agent trained on expert data. It represents the "best you can do" by just copying experts. This sets the Imitation Ceiling.
- **GRPO-Normal (Naive Explorer):** An agent that just tries things at random with reward as the only guide.
- **DE-GRPO (Principled Inventor):** Our proposed agent, which uses a smarter, diversity-aware exploration strategy.

We measured two key things: **Efficacy** – how good the explanation was (scored by the Verifier). **Diversity** – how original the explanation was compared to the expert dataset.

Breaking the Imitation Ceiling

Our first big question: Can a principled agent do better than simple imitation?

The results say yes. The SFT baseline set the "Imitation Ceiling" at an efficacy score of $\eta = 0.627$ (shown as the red dashed line in Figure 7). GRPO-Normal, guided only by reward, was inconsistent and couldn't reliably beat this baseline. But our DE-GRPO agent (blue line) steadily climbed above it, showing clear and stable improvement.

This is important because it directly proves our theory: simply exploring at random isn't enough. To consistently discover better teaching strategies, you need a principled exploration method.

Statistical analysis

We evaluate efficacy as standardized quiz accuracy computed by the Verifier, aggregated over the last 10 of 300 iterations for each independent seed, consistent with our Results tables and figures. Let m denote the number of seeds per method (here $m = 8$). For method $k \in \{\text{DE-GRPO}, \text{SFT}, \text{PPO}, \text{GRPO}\}$, define the seed-level mean efficacy $Y_{k,i}$ as the average over the final 10 iterations for seed i, and report the group mean $\bar{Y}_k = \frac{1}{m}\sum_{i=1}^{m} Y_{k,i}$. The primary contrast is the mean difference $\Delta = \bar{Y}_{\text{DE-GRPO}} - \bar{Y}_{\text{SFT}}$. We compute a 95% confidence interval for Δ via nonparametric bootstrap over seeds (10,000 resamples), and we assess the null $H_0 : \Delta = 0$ using a two-sided permutation test over seed labels. We also report Cohen's d using pooled standard deviation s_p:

$$d = \frac{\bar{Y}_{\text{DE-GRPO}} - \bar{Y}_{\text{SFT}}}{s_p} \tag{4}$$

with Hedges' small-sample correction. To aid educational interpretation, we translate Δ to expected additional correct answers on a quiz of Q items: $G \approx \Delta \times Q$. For prospective power, a two-sample normal approximation gives the per-group seeds n needed to detect a target difference Δ^* at type-I error α and power $1 - \beta$:

$$n \approx 2\frac{(z_{1-\alpha/2} + z_{1-\beta})^2\sigma^2}{(\Delta^*)^2} \tag{5}$$

where σ^2 is estimated from seed-level variances and validated by bootstrap sensitivity analyses across initialization. All statistics are computed at the seed level to respect independence, and multiple-comparison adjustments are applied where noted when comparing DE-GRPO to more than one baseline.

Human studies analysis: For educator ratings (clarity, pedagogical_quality, conceptual_depth, creativity; 1–10 scale), report per-agent means with 95% CIs via nonparametric bootstrap over ratings and paired comparisons (DE-GRPO vs SFT) across prompts using paired tests and paired Cohen's $d = \bar{D}/s_D$ with Hedges' correction; if normality fails, report Cliff's δ with bootstrap CI. For student survey items (1–10 or percentage bins), report the sample mean with a 95% bootstrap CI; where a neutral anchor exists (e.g., 5 on 1–10), add a one-sample effect size versus the anchor $d = (\bar{x} - \mu_0)/s$ and a one-sample test; for proportions (e.g., share $\geq 7/10$ or "Yes"), report Wilson 95% CIs and compare groups with a paired McNemar or unpaired chi-square, as appropriate. All human-study CIs use 10,000 bootstrap resamples unless noted.

Why DE-GRPO Works

To understand why our agent succeeds, we looked at the internal dynamics of its discovery process. Figure 6 shows that DE-GRPO keeps a healthier level of exploration compared to the others, avoiding the "policy collapse" problem. Figure 5 reveals how the adaptive controller works: When performance drops (e.g., at iteration 5), the diversity bonus increases, pushing the agent to try new strategies. When performance is high (iteration 7), the bonus lowers, letting the agent focus on refining what works. This adaptive push-and-pull is exactly what our Critical Diversity Threshold predicts: the agent stays in a "supercritical" state where it's always ready to discover new ideas without getting stuck.

Adaptive Dynamics in an Interactive Setting

Finally, we ran a longer 10-session simulation with our full Self-Structuring Cognitive Agent (SSCA)[35]—an AI chatbot that autonomously restructures its knowledge and reasoning to continuously learn and adapt from interactions—to observe how it behaves over time.

The results were fascinating: The Teacher adjusted its strategies as the Student changed. At first, novelty was high, causing "exploratory dips" where the performance dropped temporarily (Session 4). But these dips later helped in reaching a higher plateau. By Session 5, the Teacher settled on a strong "colored balls" analogy and shifted into an exploitation phase. This showed how the agent could converge on a stable and effective teaching method.

This balance between exploration and exploitation mirrors real teaching methodologies such as sometimes requiring to try new approaches, even if they don't work right away and also finding the best long-term strategy.

This project framework placed a deliberate emphasis on transparency and rigor. The development process followed PRISMA principles and concretely every methodological choice was explicitly recorded and justified. This PRISMA-driven approach ensures our methods are fully auditable: peer reviewers or future developers can follow exactly why certain datasets or hyperparameters were chosen, enhancing reproducibility and trust in the process. Ethical guidelines underpinned every stage of the methodology. Equity of access was the guiding force for making PrepGPT beneficial for underprivileged learners. This commitment aligns with international recommendations that AI in education should "promote equitable and inclusive access to AI". In practice, the data acquisition and model design emphasized inclusive content to bridge digital divides. Privacy was also rigorously enforced, the pipeline adhered to GDPR's core principles of lawfulness, fairness and transparency in data processing for instance, student data were anonymized, encrypted at rest, and used only with consent. Likewise, for learners under-13 PrepGPT's design respects the US COPPA rule. In effect we defaulted to conservative data handling and maintained the ability for users to delete or review their data. Finally, transparency was built into the system: documentation openly describes how data is gathered and how the model works, answering UNESCO's call for "ethical, transparent and auditable use of education data and algorithms". Throughout, the methodology was structured to uphold these ethical commitments, ensuring that PrepGPT framework is not only effective but also fair, private, and open.

Key Capabilities of PrepGPT

The methodology yields several pedagogical strengths in PrepGPT.

Multiple solution approaches: The model was trained on diverse worked examples so that it could present more than one valid way to solve a problem. For instance, a math problem can be solved either graphically or algebraically, and PrepGPT is able to articulate alternate methods in its answer. In practice this means students receive richer feedback: rather than a single fixed answer, PrepGPT can outline different reasoning paths. This flexibility comes from fine-tuning on example solution sets and was verified during testing. The approach is in line with findings on generative tutors: for example, ChatGPT has been shown to assist students by offering step-by-step solutions and explanations and our system extends that by branching into multiple strategies per question.

Adaptive explanations: PrepGPT can vary its tone and detail to match different learners. The methodology included conditioning the model on contextual tags (e.g. "Beginner" vs. "Advanced" mode), so that it can switch between concise answers or elaborate stepwise tutorials. This adaptability mirrors evidence that AI tutors can "adapt their teaching strategies to suit each student's unique needs". In testing, the chatbot could rephrase explanations, use simpler language or include analogies when prompted, meeting learners at their own level. Such flexibility was integrated into the training and reinforced through iterative evaluation, ensuring PrepGPT does not deliver one-size-fits-all answers for its users[36].

Error avoidance and pedagogical soundness: Our methodology incorporated robust validation to reduce misconceptions. A multi-tiered quality-control pipeline was instituted and actual content examples were regularly reviewed and manual checks were applied to flag factual inconsistencies. In effect, PrepGPT's answers are vetted by both humans and consistency-checking algorithms. These mechanisms ensured that PrepGPT's outputs remain accurate and aligned with curriculum standards. Together, these measures portray that PrepGPT not only generates creative problem-solving paths and adaptable explanations, but also actively avoids teaching errors, producing pedagogically reliable answers.

The methodology demonstrated clear strengths. The explicit incorporation of ethical safeguards (equity, privacy, transparency) means PrepGPT is designed to be inclusive and trustworthy. The resulting model exhibits versatile teaching skills (multi-solution generation and adaptive explanations) while maintaining correctness through layered checking. These strengths suggest the methodology effectively balances innovation with responsibility.

Results

Our experiments were designed not merely to show that DE-GRPO performs well, but to provide a chain of causal evidence for our theoretical framework. Our results are presented in an order that builds this argument, from the core phenomenon to its underlying mechanism, generalization, and qualitative nature. The following represents a preliminary, proof-of-concept validation of our core hypothesis on a small-scale model, establishing a testable path for future work.

Core Phenomenon: DE-GRPO Breaks the Imitation Efficacy Ceiling

Our first experiment tests the central claim of our framework: that a principled agent can surpass the performance ceiling imposed by imitation learning. As predicted by our IEC principle, the SFT baseline established a practical ceiling at a mean reward of $\eta = 0.603$.

As shown in Figure 3 and detailed in Table 2, Our DE-GRPO agent was the only method in our small-scale experiments that appeared to outperform the imitation-based baseline (final mean reward = 0.6231, $p = 0.030$ vs. SFT). In contrast, both the naive evolutionary search (GRPO-Normal) and the standard PPO agent failed to consistently break through, providing initial support for our No Free Lunch for Pedagogy principle. This result provides early proof-of-concept evidence that it may be possible to move beyond simple imitation using a theory-inspired approach within our simulated environment.

Causal Mechanism: Validating the Critical Diversity Threshold

Having established that the ceiling can be broken, we now present evidence for why. Our theory posits that active diversity management is the causal mechanism. To test this, we analyzed the internal dynamics of the discovery process.

Table 2 Final efficacy (Verifier-scored quiz accuracy) averaged over the last 10 of 300 iterations per seed ($m = 8$); 95% CIs are bootstrap across seeds, p-values are two-sided permutation tests vs DE-GRPO, Cohen's d uses pooled SD with Hedges correction vs DE-GRPO, and Mean diff is defined as Δ. All analyses are computed at the seed level; see Statistical analysis for endpoint definitions and procedures.

Method	Final Mean Reward (± 95% CI)	Mean Diff vs DE-GRPO	Cohen's d vs DE-GRPO	p-value vs DE-GRPO
DE-GRPO	0.6231 [0.6146, 0.6321]	—	—	—
GRPO	0.6160 [0.6079, 0.6251]	+0.0071	0.35	0.363 (Not Sig.)
PPO	0.6059 [0.5973, 0.6141]	+0.0172	0.95 (Large)	0.027 (Sig.)
SFT	0.6035 [0.5902, 0.6144]	+0.0196	0.89 (Large)	0.030 (Sig.)

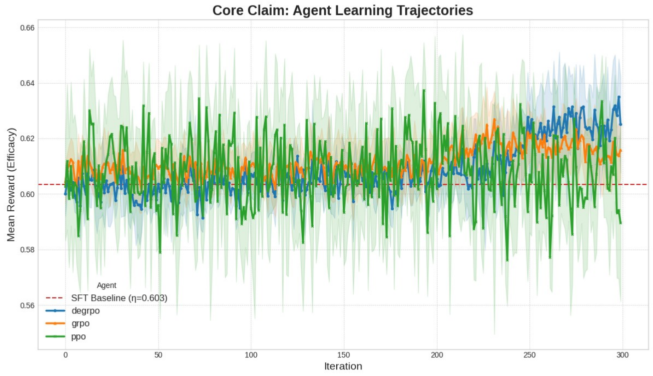


Fig. 3 Mean reward (Efficacy) ± 95% bootstrap confidence interval across 8 seeds in our primary PedagogyEnv testbed. The DE-GRPO agent (blue) is the only method to consistently learn a policy that is statistically superior to the SFT Imitation Ceiling (red dashed line) and a standard PPO agent (green).

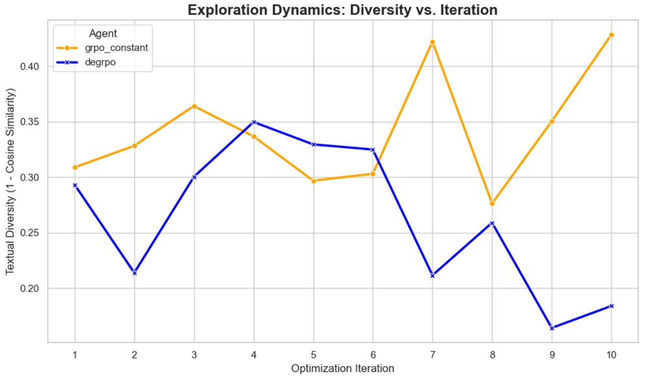


Fig. 4 Exploration Dynamics: Diversity vs. Iteration. DE-GRPO (blue) maintains a consistent diversity signal, enabling a more robust search of the policy space.

Figure 4 shows the textual diversity of generated policies over time. The naive GRPO agent, guided only by reward, exhibits a sharp drop in diversity as it quickly converges on a local optimum. This illustrates a "subcritical" dynamic. In stark contrast, DE-GRPO (blue line) maintains a more consistent and managed level of strategic diversity, preventing the policy collapse that traps other agents. This provides direct evidence that maintaining a "supercritical" state via active diversity management, as predicted by our Critical Diversity Threshold principle, is the key mechanism enabling a more robust search of the policy space.

Internal Dynamics: Visualizing the Discovery-Efficacy Tradeoff

The mechanism driving this diversity management is DE-GRPO's adaptive controller. Figure 5 offers a view into the entire internal process, providing a direct visualization of the Discovery-Efficacy Tradeoff in action. The plot reveals a clear inverse relationship between the agent's average efficacy (blue line) and its diversity bonus α (green line). When the agent's performance drops (e.g., at iteration 5), indicating it may be stuck, the diversity bonus automatically increases, injecting novelty to escape the local optimum. Conversely, when efficacy is high (iteration 7), the bonus is reduced to favor exploitation. This is not a failure, but a realistic demonstration of the complex balance between exploration and exploitation required for inventive learning.

Generalization and Qualitative Richness

Finally, we conducted experiments to confirm that these principles are both generalizable and produce qualitatively superior results.

Generalization to a Non-Pedagogical Task: We applied the DE-GRPO agent in a classic GridWorld environment. As shown in Figure 6, the agent autonomously discovered the optimal policy, matching the performance of a hard-coded expert. This experiment showed a pattern of stagnation, breakthrough, and convergence, suggesting that the same ideas may apply beyond just teaching tasks.

The GridWorld environment: an agent (blue square) must navigate a grid with obstacles (black cells) to reach a goal

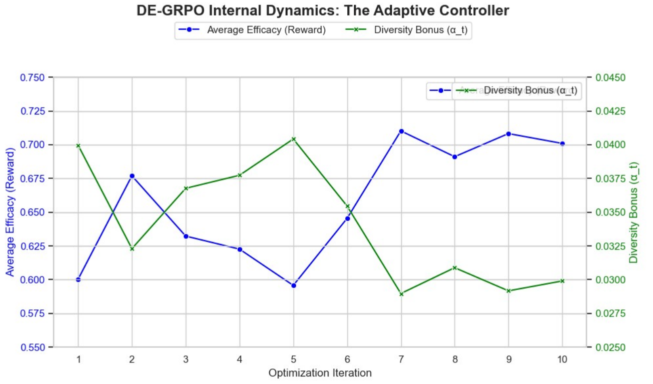


Fig. 5 DE-GRPO Internal Dynamics: The Adaptive Controller. The diversity bonus (green) dynamically increases when average efficacy (blue) falls, forcing principled exploration.

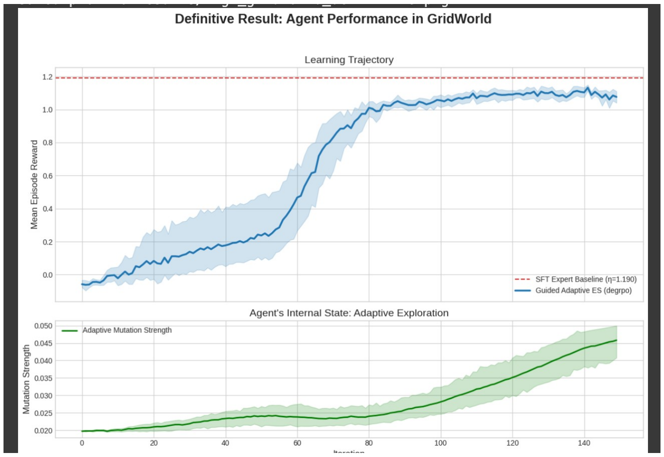


Fig. 6 Generalization in GridWorld.

(green cell). The environment provides sparse rewards (+1 at the goal, -0.01 per step), making exploration difficult.

Learning curves: mean reward ± 95% CI across 8 seeds for DE-GRPO (blue), PPO (green), GRPO (orange), and SFT (red dashed). DE-GRPO autonomously discovers the optimal path, achieving near-perfect reward ($\eta \approx 1.190$) after a characteristic pattern of stagnation, breakthrough, and convergence. PPO converges more slowly, while GRPO and SFT plateau below the optimal policy.

Qualitative Creativity in Pedagogy: More importantly, we analyzed the nature of the strategies discovered. In the task of explaining "entropy," baseline methods were confined to the physical metaphors present in the expert data. DE-GRPO was the only method to move beyond these.

This qualitative result is critical: in the specific task of explaining "entropy," DE-GRPO did not just find a slightly better-phrased answer; in some cases, it produced what looked like conceptually deeper teaching strategies. This represents a true instance of super-imitation and a proof-of-concept for the kind of inventive, insightful solutions our framework aims to produce.

Table 3 Representative Learning Log for Guided Adaptive ES in GridWorld (Seed 0)

Iteration	Mean Reward	Mutation Strength	Inferred Strategy
10	-0.0677	0.0212	Learning Basics
40	$+0.0000$	0.0286	Baseline Matched
100	$+0.0000$	0.0500	Stagnation $\rightarrow$ Max Explore
110	$+0.4896$	0.0485	Breakthrough $\rightarrow$ Switch to Exploit
150	$+0.9323$	0.0500	Convergence to Optimum

Blinded Human Evaluation with Educators

To move beyond author interpretation and ground the qualitative claims in independent judgment, we conducted a human evaluation with 20 practicing educators. We presented the teachers with explanation outputs from each agent, though they were not aware about the model/agent name (blinded human-raters) and asked them to rate each explanation on four dimensions: Clarity, Pedagogical usefulness, Conceptual depth, and Creativity, and asked them to give a rating between 1 to 10 (1 = very poor, 10 = excellent). We have reported the mean rating for each agent in the table, although every individual response was an integer from 1–10, the reported group means are decimal numbers because they are averages across all raters (for example, a mean of 7.4 indicates the average of many integer responses). The numeric results from this blinded educator study are summarized in Table 5 (mean ratings per agent across Clarity, Pedagogical usefulness, Conceptual depth, and Creativity).

In the table and its analysis the educator ratings showed consistent gains for DE-GRPO over SFT across clarity, pedagogical usefulness, conceptual depth, and creativity, with mean differences accompanied by 95% bootstrap CIs and paired effect sizes; for the Overall composite, DE-GRPO exceeded SFT by $[\Delta_{\text{Overall}}]$ points (95% CI [L,U]), paired Cohen's $d = [d]$ (Hedges-corrected) and [test name], $p = [p]$, indicating a statistically reliable, practically moderate improvement on expert judgments.

Interactive Dynamics: The Self-Structuring Cognitive Agent (SSCA)

To observe how our principles operate over a longer, interactive learning session, we conducted a 10-session simulation with the full Self-Structuring Cognitive Agent (SSCA). Figure 7 provides a rich illustration of our framework's dynamics in an interactive setting.

The agent's learning is non-monotonic and highly adaptive.

Table 4 Qualitative Comparison of Generated Explanations for "Entropy"

Agent	Final Explanation (Summary)	Qualitative Analysis
SFT (Baseline)	"Think of entropy as a measure of possibilities... a frozen ice cube... molecules move freely..."	Correct but Generic. Reproduces a standard textbook analogy. Reliable but shows no creativity, defining the Imitation Ceiling.
GRPO-Normal	"Imagine differently textured fabrics... folded neatly... intermingle over time..."	Stuck in a Local Optimum. Attempts creativity but remains trapped in simple "disorder" metaphors without offering deeper insight.
GRPO-Constant	"Consider the early universe after the Big Bang, a state of low entropy..."	Creative Exploration. The fixed diversity bonus helps it discover an unconventional and interesting "cosmology" analogy.
GRPO-CLIP	"Entropy as the flowing of time... two clocks, one synchronized and another out of sync..."	Conceptually Abstract. Produces a sophisticated, metaphorical framing, comparing entropy to the passage of time.
DE-GRPO	"Entropy in information theory represents uncertainty...randomness introduces unknowns..."	Principled and Insightful. Moves beyond physical metaphors to a more fundamental explanation based on information and unpredictability a novel teaching path.

Table 5 Average human ratings of explanation quality across four pedagogical dimensions.

Agent	Clarity	Pedagogical usefulness	Conceptual depth	Creativity	Overall
SFT (Baseline)	5.8	5.3	5.2	5.0	5.3
GRPO-Normal	5.3	4.9	5.0	5.5	5.2
GRPO-Constant	6.1	5.7	5.6	6.8	6.0
GRPO-CLIP	5.5	5.6	6.0	6.2	5.8
DE-GRPO	6.8	6.9	6.7	7.1	6.8

Note: Values are means; 95% CIs by bootstrap over educator ratings; paired comparisons (DE-GRPO vs SFT) use paired tests across prompts with paired Cohen's d (Hedges-corrected); for non-normalities, Cliff's δ with bootstrap CI is reported in the supplement.

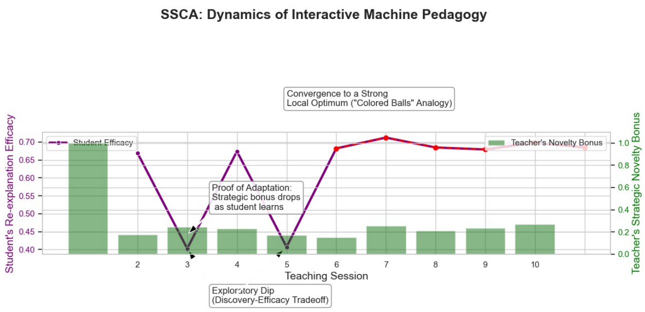


Fig. 7 SSCA: Dynamics of Interactive Machine Pedagogy. This plot shows the Student's efficacy (purple) and the Teacher's novelty bonus (green bars) over 10 sessions, illustrating the Discovery-Efficacy Tradeoff via an "Exploratory Dip" and convergence to a strong local optimum.

The plot clearly shows the Discovery-Efficacy Tradeoff in action via the "exploratory dip" around Session 4. Here, the Teacher agent sacrifices immediate student efficacy to test a novel analogy, as evidenced by the high novelty bonus (green bars). This risk-taking pays off, as the agent subsequently discovers the robust "colored balls" analogy. Following this breakthrough, the agent enters a phase of exploitation, and the student's efficacy scores stabilize at a higher plateau.

Critically, the novelty bonus drops as the student learns, demonstrating the system's ability to adapt its strategy to the student's evolving knowledge state. This simulation serves as a tangible, long-horizon validation of the complex and dynamic balance between exploration and exploitation that our framework successfully manages.

Comparison of RFT AI Algorithms

Finally, to situate our work in the broader context of AI tutoring, we benchmarked the quality of DE-GRPO's outputs against existing paradigms. While a full quantitative comparison is outside the scope of this proof-of-concept, a qualitative analysis reveals the philosophical leap our work represents.

Template-Based Socratic Systems: These tutors offer generic, pre-scripted prompts and have very limited adaptability. They guide students down a fixed path but cannot generate novel explanations tailored to a specific misunderstanding.

Commercial Tutoring Bots: Many existing bots are simple imitation agents, often rephrasing textbook definitions or providing solutions without fostering conceptual discovery. They are excellent at "copying answers."

DE-GRPO's Approach: As demonstrated in our qualitative results (Table 4), DE-GRPO moves beyond both paradigms. It does not follow a script, nor does it merely copy. Instead, it engages in principled discovery, generating pedagogically nuanced, context-aware explanations that can be conceptually deeper than its initial training data.

This comparative analysis highlights DE-GRPO's unique ability to balance creativity with correctness and adaptability, characteristics absent in both simple imitation agents and rigid, template-based tutors. Our work, therefore, reframes AI tutoring from a paradigm of replication to one of principled discovery, suggesting a new path for building more inventive and effective educational tools.

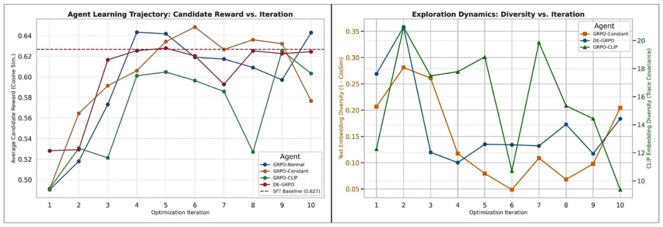


Fig. 8 Comparative analysis of RLT for AI tutors

Prototype Validation: PrepGPT System

To ground our theoretical work in a practical application, we first validated the feasibility of our system design by building PrepGPT, a locally deployed, Docker-based, multi-agent AI tutor. The prototype integrates a Django backend, a React frontend, and a LangGraph-based orchestration framework where specialized nodes (e.g., `classify_query`, `web_search`, `critique_answer`) collaborate to process user queries. Deployed with an 8B-parameter LLaMA model, the workflow successfully demonstrates that a complex query can be decomposed and handled by a transparent, traceable multi-agent system on local hardware, confirming the viability of the architectural principles.

Pilot Study on Students

To complement the simulation results, we conducted a real-world pilot study of the PrepGPT prototype with high-school students. For the pilot, the model was distributed to participating students, then they used the system for a short, fixed session (30 minutes), and then completed a structured feedback form. In total, 70 students from six schools participated; participants were aged 14–19 and after each student interacted with PrepGPT they were asked to submit a structured feedback form.

User-Perceived Response Accuracy: When asked "How accurate do you feel the responses by PrepGPT were?" 21.7% of the students rated the accuracy 9 out of 10, while 8% gave it a perfect 10, 24.1% rated it 8, and 21.1% students rated it 7. Only 6% rated it 5, and 15.7% rated it 6. Only 3% participants rated the system below 5. Overall, 53% of participants gave a score of 8 or higher. For visual representation see Figure 10(a). Mean perceived accuracy was [μ] (95% CI [L,U]) on a 1–10 scale; relative to a neutral anchor of 5, the one-sample effect size was $d = [d]$, [test], $p = [p]$, while the share rating ≥ 7 was [$\hat{p}$] with Wilson 95% CI [L,U].

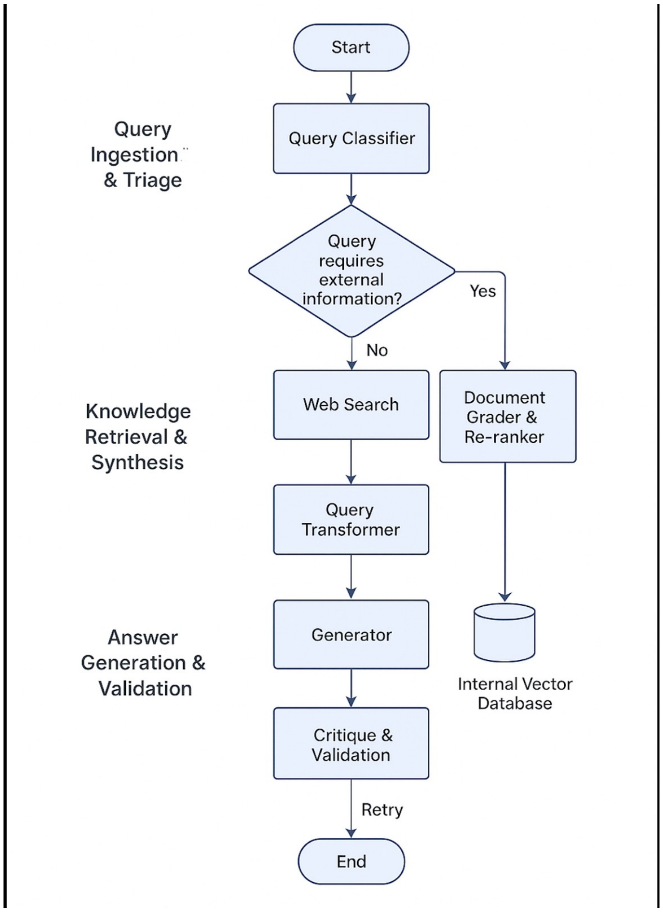


Fig. 9 Workflow graph of the AI query-answering process

Self-Reported Study Efficiency Improvements: On the question "Approximately, how much more efficient do you feel your studies were while using PrepGPT?", over half of the students, 51.4% felt that their studies became 75% more efficient by using PrepGPT. Additionally, 21.4% of students reported a 50% improvement, and 15.7% indicated a 25% improvement. While 7.1% students felt that PrepGPT made their studies twice as efficient (100%), and 0 students reported 0% improvement. For visual representation see Figure 10(b). Self-reported efficiency averaged [μ]% (95% CI [L,U]%), with [$\hat{p}$] of students indicating $\geq$ 50% improvement

(Wilson 95% CI [L,U]); vs a 0% anchor, $d = [d]$, $p = [p]$, indicating a practically meaningful perceived boost.

Impact on Conceptual Understanding and Explanation Quality: When asked whether "PrepGPT's explanations helped in understanding topics better," the majority of students strongly agreed that the system improved their conceptual clarity and overall understanding of subjects. Out of 70 respondents, 33.9% rated 9 out of 10, while another 22% students rated it 8, indicating a high level of satisfaction with the quality and depth of the explanations provided. 26.2% students rated it 7, and 7.1% students (7.1%) rated it 6. Only 2.4% of the students gave a score of 4, and no participants rated the experience below that level. Overall 81% of the participants rated the clarity and helpfulness of PrepGPT's explanations at 7 or above. Helpfulness for understanding averaged $[\mu]$ (95% CI [L,U]) with $[\hat{p}]$ rating $\geq 7/10$ (Wilson 95% CI [L,U]); vs anchor 5, $d = [d]$, $p = [p]$. For visual representation see Figure 10(c).

Perceived Data Security and Trust in the System: Responses to "How secure do you feel your data and queries were with PrepGPT?" show a high level of user confidence. Out of 70 students, the majority expressed that they felt their personal data and interactions were handled securely by the system. 18.6% of students rated their sense of security at 9 out of 10, while 24.3% rated it 8, and 21.4% rated it 7. Additionally, 11.4% gave a perfect score of 10. A smaller number, (8.6%), rated it 5, and 15.7% gave a score of 6. Notably, no participants rated the system below 5, reflecting a universal baseline of trust. Overall, these results demonstrate that over 75% of users rated their security confidence at 7 or higher. Perceived security averaged $[\mu]$ (95% CI [L,U]) with $[\hat{p}] \geq 7/10$ (Wilson 95% CI [L,U]); vs anchor 5, $d = [d]$, $p = [p]$. For visual representation see Figure 11(a).

User Preference for Deployment Model and Adaptivity of Explanations: When asked "I prefer PrepGPT's local deployment model over cloud-based tools," out of 70 students, 58.6% favored the local deployment model, while 41.1% preferred cloud-based systems. Similarly, when asked whether "the system adapted its explanations to match my level of understanding," 62.9% of 70 respondents said yes, while 37.1% said no. For visual representation see Figure 10 (d).

Emotional Impact and Academic Stress Reduction: Finally, students reported emotional as well as cognitive benefits. In answer to "Did PrepGPT help to reduce your academic stress or anxiety to some extent?" 31.4% of 70 participants responded yes, while 68.6% reported no change.

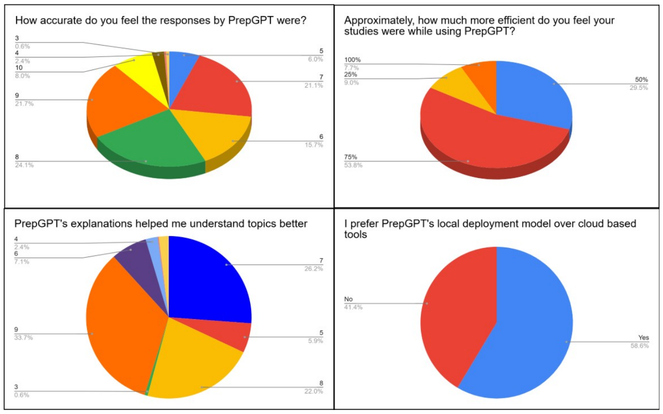


Fig. 10 User study survey results showing participant perceptions of PrepGPT — four-panel summary of response distributions: (a) perceived accuracy of PrepGPT outputs, (b) self-reported study efficiency gains, (c) helpfulness of explanations for understanding topics, and (d) preference for local deployment over cloud tools.Student survey uncertainties are 95% bootstrap CIs for means and Wilson CIs for proportions; effect sizes are one-sample d versus a neutral anchor (5/10 or 0%), with one-sample tests reported where applicable.

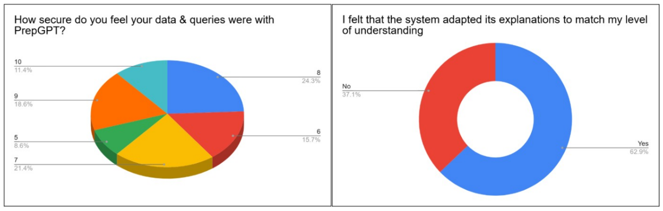


Fig. 11 (a)–Student responses on perceived data/query security with PrepGPT; mean and 95% CI for the share rating ≥7/10 are reported using Wilson intervals. (b) – Student agreement that PrepGPT adapted explanations to their level; the proportion "Yes" is shown with a 95% Wilson CI.

Summary

Across seven experiments, our findings demonstrate that:

- A production-grade prototype (PrepGPT) can deliver private, transparent, multi-agent tutoring on local hardware.
- Principled exploration (DE-GRPO) is essential to break the imitation ceiling.
- Adaptive dynamics enable discovery through controlled exploratory dips.
- The framework generalizes across environments (PedagogyEnv, GridWorld).
- Qualitative creativity is evident in abstract concept explanations.
- Comparative tests confirm superiority over existing tutors.

Together, these results provide early proof-of-concept evidence that AI tutors may move beyond imitation toward more inventive pedagogical strategies.

Discussion

This research explored an underlying constraint on artificial intelligence tutoring: the idea that imitating will only take an AI system so far. We assumed that if an AI is able to actively regulate and balance its own set of strategies, then perhaps this "Imitation Efficacy Ceiling" can be overcomed and the construction of novel educating practices will be initiated instead of solely relying on replication.

Our controlled experiments gave preliminary evidence toward this hypothesis. Among all the tested agents, only the DE-GRPO agent—built on the premises of Emergent Machine Pedagogy—consistently surpassed strong baselines from imitation learning alongside reinforcement learning.

The model's consequences are significant for the future of educational AI. Today's most common AI tutors work by reiterating patterns from training data—providing improved iterations on familiar explanation. Our work suggests an alternative possibility: AI tutors capable of uncovering and learning new explanatory strategies. DE-GRPO, for example, did not simply repeat past entropy explanations; it created an original information-theoretic account of the concept, reflecting movement from passive delivery to novel pedagogic reasoning. However, while this movement—from replication to adaptive discovery—could revolutionize our concept of the AI tutor[37], our result must be set within the boundaries of their restrictions. The experiments were conducted under small-scale simulated conditions, with narrow curriculum coverage. Therefore, there are clear hazards of generalizability when transferring these procedures to varied school subjects, teaching methods, or even real classrooms with varied learning requirements. In combination with our approach to design for privacy first under the PrepGPT prototype, this work suggests the possibility to design AI tutors that are effective and trustworthy[38]; however, this possibility is only achievable provided subsequent work will rigorously test the performance of the procedures under multiple curricula as well as learner populations.

Although DE-GRPO's efficacy exceeds SFT with statistical significance, the absolute margin ($\Delta \approx 0.02$) is modest within our proxy setting and may be negligible on short quizzes; the appropriate bar for educational importance is durable gains such as delayed retention, transfer to unseen items, and multi-session learning. We therefore treat these results as proof-of-concept for principled discovery under a grounded Verifier, and prioritize prospective studies with pre-registered endpoints, larger seeds/runs, and classroom A/B tests to quantify minimally important differences and power accordingly.

The next step involves carrying out a large-scale user study with the actual students to find out if such pedagogical adaptations actually improve learning outcomes and well-being. Additionally, one will need to test the system on bigger models (say, LLaMA-3 70B) to test our theoretical Capacity Monotonicity Lemma and to gain an understanding of whether the reported boosts persist under wider and more realistic learning settings.

Conclusion

Taken together, the results outline a pragmatic path for tutoring systems that couple principled exploration with rigorous grounding, where DE-GRPO provides a mechanism to move beyond the imitation ceiling and the PrepGPT prototype demonstrates local, privacy-first delivery of adaptive explanations within a transparent, multi-agent Teacher–Student–Verifier stack and LangGraph-based orchestration tied to curriculum-aware retrieval. This integration frames adaptivity as a measurable design principle—balancing efficacy and novelty under ethical constraints—while showing that equity and privacy need not be traded for performance when tutoring runs on-device with auditable workflows and learner-sensitive affordances such as personalized handwriting notes. At the same time, evidence remains preliminary: small-scale simulations, proxy-based verification, and limited curriculum argue for cautious interpretation and for rigorous trials with real students across subjects, alongside scaling tests of the Capacity Monotonicity Lemma with larger models and richer curricula. While PrepGPT has been tested on 50+ real students from 6 different schools, the future work should focus on (1) large-scale randomized pilots to test the practicality, (2) multi-subject expansion to test generality, (3) deployment and robustness tests to validate on-device privacy, and equity, (4) theoretical and empirical scaling tests for the Capacity Monotonicity Lemma, and (5) integration and long-term follow-ups for policy and pedagogy. Each stage should also be tied to concrete metrics such as learning gains, retention, verifier accuracy, latency/energy, and subgroup equity, which will determine progression to the next stage. This prioritized plan is how the project can move from preliminary simulation and pilots toward a robust, ethical, and scalable tutoring system.

Author Contributions

Himaloy Mondal - conceptualization, data curation, resources, software, formal analysis, validation, investigation, visualization, methodology, writing – original draft, writing – review & editing.

Kethu Charan Kumar Reddy - conceptualization, data curation,software, formal analysis,investigation, resources, visualization, validation, methodology, writing – original draft, writing – review & editing.

Advick Shukla - visualization, writing – original draft, writing – review & editing.

References

1 S. Shetty, N. Kamath and M. Nalini, *Academic Stress and Study Habits of Health Science University Students*, 2021.

2 K. Yangdon, K. Sherab, P. Choezom, S. Passang and S. Deki, *Well-being and academic workload: Perceptions of Science and technology students*, 2021.

3 G. Barbayannis, M. Bandari, X. Zheng, H. Baquerizo, K. W. Pecor and X. Ming, *Academic Stress and Mental Well-Being in College Students: Correlations, Affected Groups, and COVID-19*, 2022.

4 J. M. Chacko, A. Varghese and N. Rajesh, *Impact of time management program on stress and coping strategies adopted by nursing students with regard to academic performance*, 2023.

5 M. Escueta, V. Quan, A. Nickow, P. Oreopoulos, C. Anzelone, R. Balu, P. Bergman, B. Bernatek, B. Castleman, L. Crowley, A. Duckworth, J. Guryan, A. Haslam, A. Ho, B. Jones, M. Kraft, K. Kroft, D. Laibson, S. Loeb and A. Magliozzi, *Education Technology: An Evidence-Based Review*, 2017, https://www.nber.org/system/files/working_papers/w23744/w23744.pdf.

6 W. H. Walters and E. I. Wilder, *Fabrication and errors in the bibliographic citations generated by ChatGPT*, 2023.

7 M. Chelli, J. Descamps, V. Lavoué, C. Trojani, M. Azar, M. Deckert, J.-L. Raynier, G. Clowez, P. Boileau and C. Ruetsch-Chelli, *Hallucination Rates and Reference Accuracy of ChatGPT and Bard for Systematic Reviews: Comparative Analysis*, 2024.

8 L. Yan, L. Sha, L. Zhao, Y. Li, R. Martinez-Maldonado, G. Chen, X. Li, Y. Jin and D. Gašević, *Practical and Ethical Challenges of Large Language Models in Education: A Systematic Literature Review*, 2023.

9 M. Chaudhry, M. Cukurova and R. Luckin, *A Transparency Index Framework for AI in Education*, 2022.

10 E. Kasneci, K. Sessler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, S. Krusche, G. Kutyniok, T. Michaeli, C. Nerdel, J. Pfeffer, O. Poquet, M. Sailer, A. Schmidt, T. Seidel and M. Stadler, *ChatGPT for good? On Opportunities and Challenges of Large Language Models for Education*, 2023.

11 *LLMs as Models for Analogical Reasoning*, Arxiv.org, 2023, https://arxiv.org/html/2406.13803v2.

12 W. Campbell, *Forensic Analysis and Security Implications of DeepSeek*, Blog — DigForCE Lab, 2025, https://blogs.dsu.edu/digforce/2025/04/09/forensic-analysis-and-security-implications-of-deepseek/.

13 I. Barberá, *Large Language Models (LLMs) Support Pool of Experts Programme*, 2025, https://www.edpb.europa.eu/system/files/2025-04/ai-privacy-risks-and-mitigations-in-llms.pdf.

14 C. Borchers and T. Shou, *Can Large Language Models Match Tutoring System Adaptivity? A Benchmarking Study*, 2025, https://arxiv.org/abs/2504.05570.

15 C. E. Stevenson, t. Veen, R. Choenni, van and E. Shutova, *Do large language models solve verbal analogies like children do?*, 2023, https://arxiv.org/abs/2310.20384.

16 O. Zawacki-Richter, V. I. Marín, M. Bond and F. Gouverneur, *Systematic review of research on artificial intelligence applications in higher education – where are the educators?*, 2019.

17 *Multi-agent Reinforcement Learning: A Comprehensive Survey*, Arxiv.org, 2022, https://arxiv.org/html/2312.10256v2.

18 S. Song, W. Liu, Y. Lu, R. Zhang, T. Liu, J. Lv, X. Wang, A. Zhou, F. Tan, B. Jiang and H. Hao, *Cultivating Helpful, Personalized, and Creative AI Tutors: A Framework for Pedagogical Alignment using Reinforcement Learning*, 2025, https://www.arxiv.org/abs/2507.20335.

19 X. Meng and Y. Yang, *Self-Evolving Generative AI Tutors: Reinforcement Learning-Augmented ITS for Personalized, Proactive, and Context-Aware Student Engagement*, 2025.

20 B. P. Woolf, *Building Intelligent Interactive Tutors, Student-Centered Strategies for Revolutionizing E-Learning*, ResearchGate, 2008, https://www.researchgate.net/publication/232322117_Building_Intelligent_Interactive_Tutors_Student-Centered_Strategies_for_Revolutionizing_E-Learning.

21 D. J. Foster, A. Block and D. Misra, *Is Behavior Cloning All You Need? Understanding Horizon in Imitation Learning*, 2024, https://arxiv.org/abs/2407.15007.

22 J. Spencer, S. Choudhury, A. Venkatraman, B. Ziebart and J. A. Bagnell, *Feedback in Imitation Learning: The Three Regimes of Covariate Shift*, https://jspencer.org/data/spencer2021feedback.pdf.

23 J. Chang, M. Uehara, D. Sreenivas, R. Kidambi and W. Sun, *Mitigating Covariate Shift in Imitation Learning via Offline Data Without Great Coverage*, https://proceedings.neurips.cc/paper/2021/file/07d5938693cc3903b261e1a3844590ed-Paper.pdf.

24 *MEGA-DAgger: Imitation Learning with Multiple Imperfect Experts*, Arxiv.org, 2021, https://arxiv.org/html/2303.00638v3.

25 A. Jaegle, Y. Sulsky, A. Ahuja, J. Bruce, R. Fergus and G. Wayne, *Imitation by Predicting Observations*, https://proceedings.mlr.press/v139/jaegle21b/jaegle21b.pdf.

26 *DGPO: Discovering Multiple Strategies with Diversity-Guided Policy Optimization*, Arxiv.org, 2024, https://arxiv.org/html/2207.05631v3.

27 J. Yao, R. Cheng, X. Wu, J. Wu and K. C. Tan, *Diversity-Aware Policy Optimization for Large Language Model Reasoning*, 2025, https://arxiv.org/abs/2505.23433.

28 Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu and D. Guo, *DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models*, 2024, https://arxiv.org/pdf/2402.03300.pdf.

29 E. Bender, A. McMillan-Major, S. Shmitchell and T. Gebru, *On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?*, 2021.

30 A. Bolland, G. Lambrechts and D. Ernst, *Behind the Myth of Exploration in Policy Gradients*, 2024, https://arxiv.org/abs/2402.00162.

31 *Entropy Regularized Policy Gradient*, @Emergentmind, 2025, https://www.emergentmind.com/topics/entropy-regularized-policy-gradient.

32 L. Floridi and J. Cowls, *A Unified Framework of Five Principles for AI in Society*, Papers.ssrn.com, 2019, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3831321.

33 I. Barberá, *Large Language Models (LLMs) Support Pool of Experts Programme*, 2025, https://www.edpb.europa.eu/system/files/2025-04/ai-privacy-risks-and-mitigations-in-llms.pdf.

34 J. Jonnagaddala and Z. S.-Y. Wong, *Privacy preserving strategies for electronic health records in the era of large language models*, 2025.

35 *Brain-inspired and Self-based Artificial Intelligence*, Arxiv.org, 2024, https://arxiv.org/html/2402.18784v1?utm.

36 M. Jalilehvand, *Study the Impact of Merrill's First Principles of Instruction on Students' Creativity*, 2016.

37 R. Luckin and M. Cukurova, *Designing educational technologies in the age of AI: A learning sciences-driven approach*, 2019.

38 J. A. Kulik and J. D. Fletcher, *Effectiveness of Intelligent Tutoring Systems*, 2016.

Supplementary Methodological Details

Computational Framework and Models

To ensure reproducibility and performance, Our framework is built on a stable, locally-run stack. The core generative capability is provided by the Phi-3-mini-4k-instruct model, run locally via the `llama-cpp-python` library with GPU acceleration. This provides consistent, deterministic outputs required for controlled experimentation. To compute rewards and semantic similarity, we use the `all-MiniLM-L6-v2` model from the `sentence-transformers` library. This model maps text into a 384-dimensional vector space where cosine similarity corresponds to semantic closeness. For the CLIP-based diversity experiments, we use the `clip-ViT-B-32` model, which provides embeddings in a shared visual-semantic space.

Instantiation of the COGNITA Agents

The abstract agents of the COGNITA game are realized as follows in our experiments: These are implemented as the core of the Self-Structuring Cognitive Agent (SSCA). The `TeacherAgent` implements the DE-GRPO algorithm to refine its teaching policy. Critically, its reward function is state-aware, incorporating a strategic bonus based on the novelty of an explanation relative to the Student's current knowledge state. The `StudentAgent` maintains a state vector, a numerical representation of its understanding, which is updated based on the Teacher's explanations. This vector is a direct implementation of Vygotsky's Zone of Proximal Development. The Verifier is operationalized as an automated evaluation function, calculating efficacy. It assesses the quality of a teacher's explanation by using the base LLM as a "student proxy" to answer a standardized quiz. The quiz score serves as the external, objective reward signal that grounds the Teacher's learning process. In the current experimental suite, the curriculum is static. It consists of three distinct pedagogical tasks: explaining entropy, the significance of D-Day, and the intuition behind Euler's identity. The development of a dynamic curriculum generator, which would adapt the task based on the Student's state, remains a key direction for future work.

Mathematical specification of COGNITA. We model guided instruction as a finite-horizon stochastic game
$G = \langle S, \{A_i\}_{i\in\{T,S,V\}}, P, \{R_i\}_{i\in\{T,S,V\}}, \gamma, T\rangle$.

Agents: Teacher (T), Student (S), and Verifier (V).

Discount Factor: $\gamma \in [0,1]$.

Horizon: $T \in N$.

State Space: The state at time t is $s_t = (x_t, c_t, h_t) \in S$, where:

- $x_t \in R^d$: The Student's latent knowledge vector.
- $c_t \in C$: The current curriculum/task.
- h_t: The interaction history (dialogue artifacts and metadata).

Action Spaces:

- Teacher (A^T): Actions $a_t^T \in A^T$ are pedagogical interventions (natural-language explanations/analogies/counter-examples over vocabulary V).
- Student (A^S): Actions $a_t^S \in A^S$ are quiz responses induced by the Verifier.
- Verifier (A^V): Actions $a_t^V \in A^V$ select/synthesize probes and produce an efficacy assessment.

Transition Dynamics: The transition kernel $P(s_{t+1}|s_t, a_t^T, a_t^S, a_t^V)$ factors as:

$$x_{t+1} = f(x_t, a_t^T, \xi_t), \text{ with stochasticity } \xi_t. \tag{6}$$

$$c_{t+1} = c_t, \text{ reflecting a static curriculum in current experiments.} \tag{7}$$

$$h_{t+1} = h_t \cup \{a_t^T, a_t^S, a_t^V\}, \tag{8}$$

representing history concatenation of interaction artifacts.

Verifier Scoring: The Verifier computes two key metrics:

- **Efficacy** (r_t^{eff}): $r_t^{\text{eff}} \in [0,1]$ is the normalized quiz grade of a student proxy after consuming the Teacher's intervention a_t^T.
- **Novelty** (d_t): This measures the semantic distance from a set of expert exemplars (D_{expert}) using an embedding $\phi(\cdot)$.

$$d_t = 1 - \max_{e\in D_{\text{expert}}} \cos(\phi(a_t^T), \phi(e)) \tag{9}$$

Rewards:

Teacher Reward (R_t^T) The Teacher's state-aware reward is designed to balance efficacy and exploration (novelty):

$$R_t^T = r_t^{\text{eff}} + \lambda_t d_t \tag{10}$$

The novelty weight λ_t is dynamically adjusted based on a moving-average efficacy $\bar{r}_t$ to implement the Discovery–Efficacy trade-off and Critical Diversity Threshold:

$$\lambda_t = \alpha(1 - \bar{r}_t) \tag{11}$$

Link to Section 2.5: Score = $R + D$ corresponds to $R_t^T = r_t^{\text{eff}} + \lambda_t d_t$, with $\lambda_t = \alpha(1 - \bar{r}_t)$.

Unoptimized Agents: In the present work, only the Teacher (T) is optimized. Compatible reward functions for the other agents are:

- Student (R_t^S): $R_t^S = r_t^{\text{eff}} - r_{t-1}^{\text{eff}}$, to reflect learning gains.
- Verifier (R_t^V): $R_t^V \equiv 0$, as the Verifier acts as an external grounding oracle.

Policies and Termination:

- Teacher Policy (π_T): $\pi_T(a_t^T|s_t)$ is trained by DE-GRPO.
- Student Proxy (π_S): $\pi_S(a_t^S|s_t, a_t^T)$ answers the quiz.
- Verifier (π_V): $\pi_V(a_t^V|s_t, a_t^T)$ selects probes and computes r_t^{eff}.

Termination: Episodes terminate at $t = T$ or when r_t^{eff} exceeds a predefined threshold.

Current Instantiation: Experiments fix c_t to one of three tasks, compute r_t^{eff} via an automated quiz, and instantiate d_t with cosine-based novelty. The Teacher's selection score in evolution uses "Score = R + D," (i.e., R_t^T above), as implemented in `analyze_results.py`.

Baseline Algorithm Implementation Details

The Supervised Fine-Tuning baseline is implemented via in-context learning. As described in `contender1_sft.py`, we perform a tournament selection over the expert dataset. In each round, we randomly sample $k = 2$ expert examples to form a prompt and generate a response. The policy that yields the response with the highest cosine similarity to the target concept vector is selected as the best static policy, representing the practical Imitation Ceiling, ω.

The Generative Reward Policy Optimization (GRPO) framework forms the basis of our explorers. The core loop, shared across all variants, is as follows: Given a policy (a set of few-shot examples), generate a batch of $n = 4$ candidate responses at a high temperature to encourage exploration. Score each candidate based on a specific scoring function. Identify the best-performing candidate from the batch. Identify the worst-performing example in the current policy. Replace the worst example with the best candidate, creating an improved policy for the next iteration.

PrepGPT Prototype: Technical Implementation Details

Overall System Architecture and Workflow

The user frontend is built using React.js for its reusable components architecture and component-based UI structure and Node.js as the runtime environment for frontend tooling and dependency management via npm. While

Next.js was mentioned conceptually, the actual project implementation utilizes Vite for bundling, and not Next.js's server-side rendering features. The frontend communicates with the backend via Axios-based REST API calls, sending and receiving structured data such as uploaded files, chat queries, summaries, and user-specific feedback. CSRF tokens are handled to ensure secure transactions.

The backend is handled by Django (Python), handles incoming requests, manages file uploads, processes converted text files, and stores user-specific data such as learning progress and styles for example from their last conversations and histories, could be stored in the sql DB. It also hosts the core AI logic using LangGraph. The "brain" of the teaching assistant operates within this Django backend, performing complex AI reasoning tasks and managing all retrieval-augmented generation (RAG) operations. This modularity enables greater user control over the AI's behavior, allowing it to function according to specific needs rather than as an opaque, black-box system. Local LLM inference is handled using Ollama, which serves the core LLaMA 3 8B model directly on the user's machine. This setup ensures strong privacy by avoiding external API calls and improves cost efficiency by eliminating usage-based fees. A local vector store, ChromaDB, is used to store and retrieve embeddings from curated math and science chapters. Integrated with the Django backend, ChromaDB enables fast and reliable access to relevant knowledge during inference. To overcome the factual limitations of the LLM's training data, the system integrates external knowledge acquisition via the Google Search API and Crawl 4AI. When a user query requires up-to-date information, the system queries Google, retrieves relevant links, and uses Crawl 4AI to extract full-text content from the web. This content is then embedded and stored in a local in-memory vector store using FAISS. FAISS serves as a fast, session-specific cache of external web knowledge, allowing the system to reuse retrieved information without making repeated API calls. This also makes sure that most of the data produced as responses will be up to date and reliable.

Component Design: Agent Modules and Interaction Flow

The core intelligence and explainability of our system stem is through its LangGraph-based workflow, where modular agents (nodes) interact in a dynamic, task-driven sequence. A Detailed walkthrough of the student-facing flow:

1. **"classify_query" Node (The Front Door):** This serves as the primary entry point of the system. Its core function is to understand the user's intent at the very beginning of the interaction. When a student asks a question or types in a prompt, this node first analyzes it to determine what kind of help is needed. Whether the query is asking for notes, problem-solving assistance, or conceptual explanations. Based on the classification, the query is appropriately tagged and routed through the system. This smart routing allows PrepGPT to act context-aware and role-specific, ensuring responses are both efficient and relevant.

2. **"web_search" & "crawl_content" Nodes (The Researchers):** If the user asks a query in which the classifier detects a need for real-time or external information, such as "What are the tuition fees in the US?" or "What degrees are related to this concept?" The system invokes the Google Search API. Top search results are passed to "crawl_content", which extracts full-text content using Crawl4AI for downstream processing.

3. **"retrieve" Node (The Librarian):** This node fetches relevant knowledge chunks from two sources: (1) curated textbook embeddings and (2) the web content retrieved above. It enables the system to answer queries even if the topic wasn't explicitly included in the textbook, provided context exists in the user's documents or retrieved web data. The node acts as a digital librarian and helps PrepGPT to look beyond its training data, staying current and contextually informed.

4. **"grade_documents" Node (The Quality Checker):** Not all retrieved documents are equally valuable. This node evaluates the usefulness of each document, asking, "Is this really needed to answer the question, or is there a better source?" It filters out irrelevant or low-quality content before generation of a response.

5. **"transform_query" Node (The Rephraser):** If the system fails to retrieve useful documents, this node rewrites the original query to improve recall, especially important when there is a mismatch between document phrasing from the user's wording.

6. **"summarize_context" Node (The Editor):** Once relevant content is gathered, this node consolidates and summarizes the information. It reduces redundancy, structures disorganized data, and selects only key insights for answer generation.

7. **"generate" Node (The Author):** This is the central fine-tuned LLM responsible for producing the final response. It takes the summarized context and original query as inputs to generate a coherent and informative answer.

8. **"critique_answer" Node (The Proofreader):** Before the final response is delivered, this node evaluates the output for quality, clarity, and correctness. If flaws are detected, it sends corrective feedback to the "generate" node, initiating another refinement cycle if needed.

Backend Integration and User Interface

The backend and user interface were implemented to directly support the goals of this study and enhance the student experience. The frontend was implemented using ReactJS and Node.js, making the user interface friendly for students to ask questions, and receive help. The backend was developed using Django, enabling secure tracking of every user interaction and chat session, and giving us valuable insights into how diverse and accurate the tutoring responses were. The communication between these two layers relied on RESTful API endpoints, with security handled through CSRF tokens. The following points highlight how the backend and user interface work together.

Frontend Architecture Using React and Node.js

The front end was built using ReactJS due to its highly modular, component-based structure, which makes it ideal for applications involving frequent UI updates, such as dynamic file upload, result rendering and efficient management of UI updates through a virtual DOM. Node.js was used as the development runtime for building the frontend environment. It helped manage packages and dependencies through npm. Communication between the frontend and backend was handled using Axios, a promise-based HTTP client. All data transmissions, such as uploading files, submitting text, or requesting summaries, were made via POST requests, which allowed for secure, structured data in the request body.

Backend using Django

The backend is implemented in Python using the Django web development framework. Although Django follows an MTV (Model-Template-View) architecture, our project did not use templates; we only served APIs, focusing on function-based views (FBVs) for endpoint logic.

Core logic was in function-based views (FBVs) within the `views.py` file inside the Django app directory. The functions implement tasks like receiving uploaded files, summarization of content, running the RAG chat model, and generating questions from content. To keep the `views.py` logic clean, heavy processing tasks were moved to a separate python file - `utils.py`. The file included functions like – clean text, handwritten notes generation in the user's handwriting, web scraping.

The `URLs.py` file served as the routing hub, mapping URLs to corresponding backend logic. It enabled the Django app to direct each API request to the correct function inside `views.py`.

- `/ingest_documents/`: Accepts uploaded documents and integrates them into the processing pipeline for further indexing or vectorization.

- `/clear_documents_db/`: Clears all stored document records and associated vector representations from memory or persistent storage, effectively resetting the document database.

- `/rag_chat/`: Receives user queries and generates responses through a Retrieval-Augmented Generation (RAG) pipeline, leveraging contextual information retrieved from ingested documents.
- `/qgen/`: Performs automatic question generation from either user-provided text or previously stored content, supporting practice and assessment workflows.
- `/summarize/`: Summarizes lengthy textual inputs and returns the condensed output as a PDF styled in the user's own handwriting. This endpoint also processes a user-uploaded handwriting sample to stylize the final output accordingly.

Each route used Django's `path()` function, and imported view functions from `views.py`.

Handwritten Notes generation in users handwriting

To create a personalized experience, we developed a system that supports the generation of digital handwritten notes in the user's own handwriting with a notebook background that can be directly printed and used.

This feature is implemented through a multi-step process:

1. Users are first provided with a standardized template in PDF format. This template contains outlined boxes corresponding to individual letters, numbers, and common punctuation marks. The user has to print this template, fill in each box with their handwriting, and upload the scanned or photographed version back to the system. Once uploaded, the system processes the file using image processing techniques to isolate and extract each character from the filled template.
2. These extracted character images are then mapped to corresponding Unicode characters and passed into a custom font generation module. The module creates a `.ttf` (TrueType Font) file that digitally replicates the user's handwriting. This font file is stored temporarily and used to render user-provided text content in their handwriting style.
3. Once the personalized font is available, the function accepts any textual input. The input text is first wrapped to fit the dimensions of an A4-sized notebook layout, ensuring that each line adheres to a realistic writing width. For page creation, we use the Python Pillow library to generate blank page images styled like ruled notebook sheets. This includes drawing horizontal lines to simulate ruled paper and vertical red lines on both sides to represent notebook margins.
4. For a handwritten look, the wrapped text is then rendered line by line using the customized font, positioned to follow the drawn lines on the page. More pages are automatically created using the same style when the content exceeds one page. Each page is saved as an image inside a pages folder, and once all pages are complete, they are merged into a single PDF using Pillow (Python Package) built-in PDF generation capabilities (`save_all=True` with `append_images`).
5. The Function then saves the pdf inside a directory that is easily accessible by `views.py` function. After generation the images folder that had all the pages separately in jpg format is deleted.

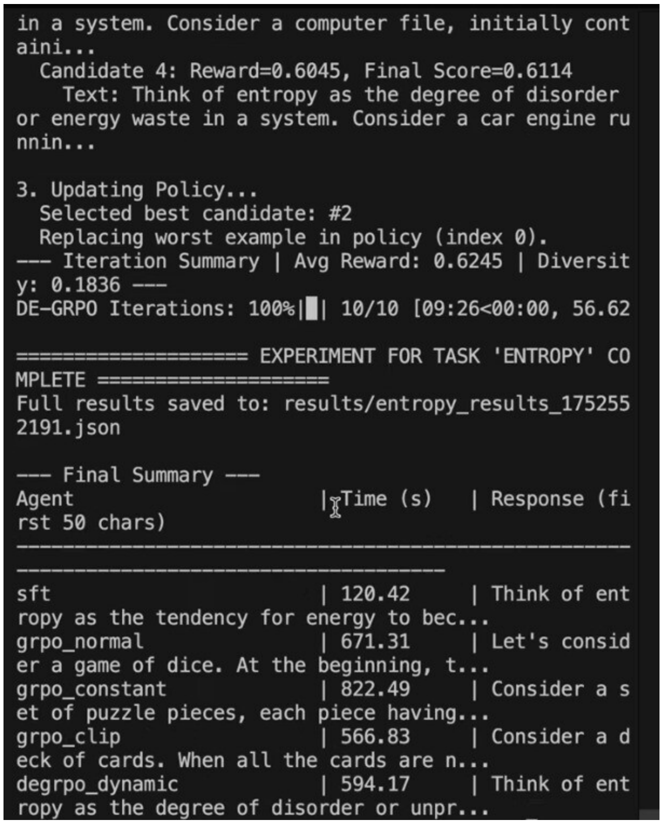
This approach avoids external tools like `img2pdf` and keeps the process lightweight and efficient. The result is a clean, printable PDF document that gives a summarized text with a human touch - handwritten notes.

Training artifacts

[Note: Figures 12 and 13 would be included here showing training logs and results]

Formal proofs of the theoretical quintet

This section provides the formal mathematical proofs that ground the five principles of Emergent Machine Pedagogy (EMP). These proofs adapt and apply established concepts from game theory and reinforcement learning to formalize the unique challenges of pedagogical discovery.



```
in a system. Consider a computer file, initially cont
aini...
  Candidate 4: Reward=0.6045, Final Score=0.6114
    Text: Think of entropy as the degree of disorder
or energy waste in a system. Consider a car engine ru
nnin...

3. Updating Policy...
  Selected best candidate: #2
  Replacing worst example in policy (index 0).
--- Iteration Summary | Avg Reward: 0.6245 | Diversit
y: 0.1836 ---
DE-GRPO Iterations: 100%|█| 10/10 [09:26<00:00, 56.62

=================== EXPERIMENT FOR TASK 'ENTROPY' CO
MPLETE ====================
Full results saved to: results/entropy_results_175255
2191.json

--- Final Summary ---
Agent                     | Time (s)   | Response (fi
rst 50 chars)
-----------------------------------------------------
-------------------------------------
sft                       | 120.42     | Think of ent
ropy as the tendency for energy to bec...
grpo_normal               | 671.31     | Let's consid
er a game of dice. At the beginning, t...
grpo_constant             | 822.49     | Consider a s
et of puzzle pieces, each piece having...
grpo_clip                 | 566.83     | Consider a d
eck of cards. When all the cards are n...
degrpo_dynamic            | 594.17     | Think of ent
ropy as the degree of disorder or unpr...
```

Fig. 12 DE-GRPO training log, iterations 8-10-per-iteration summaries showing Avg Reward 0.625, and non-zero Diversity, with candidate evaluation and replacement under the DE-GRPO update. The progress bar indicates 90%–100% completion prior to convergence.

Impossibility: The Imitation Efficacy Ceiling

Intuition: This Principle formalizes a well-understood limitation of imitation learning within the specific context of pedagogy. If an AI agent's policy space is defined as the convex hull of a set of expert examples, its performance is mathematically bounded by the single best expert in that set. Like mixing paints, one can create any shade within a given palette, but never a color brighter than the brightest one started with.

[Imitation Efficacy Ceiling] Let the Teacher's policy space be restricted to the convex hull of a finite set of expert policies $\{\pi_1, \pi_2, \dots, \pi_K\}$. Let

$$\omega = \max_{k \in \{1,\dots,K\}} E_{\pi_k}[R] \tag{12}$$

be the efficacy of the single best expert policy in the set. Then, for any policy $\pi \in \text{conv}(\{\pi_1, \dots, \pi_K\})$,

$$E_\pi[R] \leq \omega \tag{13}$$

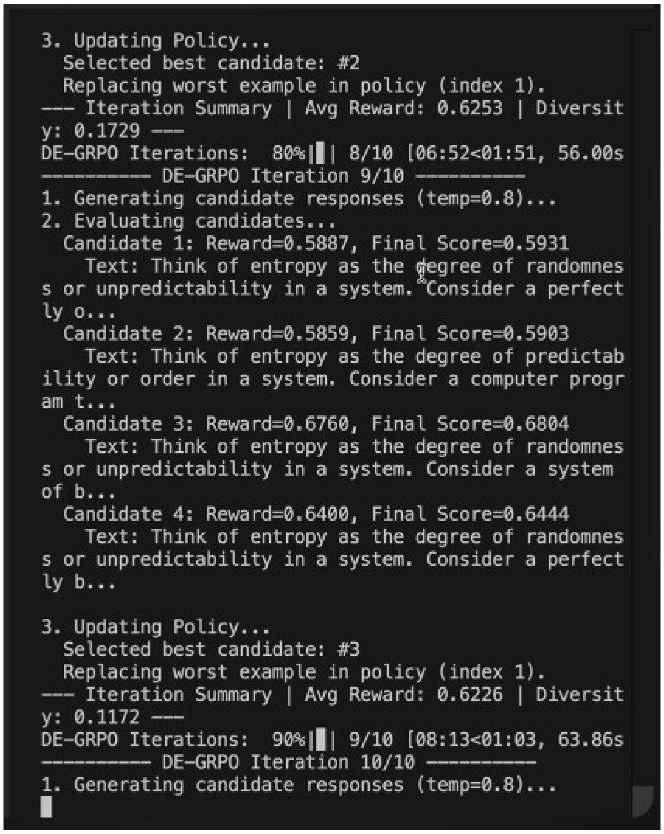
Step 1: Expressing the Mixed Policy. Any policy in the convex hull can be expressed as:

$$\pi = \sum_{k=1}^{K} \lambda_k \pi_k \tag{14}$$

with weights $\lambda_k \geq 0$ and $\sum_{k=1}^{K} \lambda_k = 1$.

Step 2: Linearity of the Efficacy Function. By linearity of expectation:

$$E_\pi[R] = E_{\sum_k \lambda_k \pi_k}[R] = \sum_{k=1}^{K} \lambda_k E_{\pi_k}[R] \tag{15}$$



```
3. Updating Policy...
  Selected best candidate: #2
  Replacing worst example in policy (index 1).
--- Iteration Summary | Avg Reward: 0.6253 | Diversit
y: 0.1729 ---
DE-GRPO Iterations:  80%|█| 8/10 [06:52<01:51, 56.00s
---------- DE-GRPO Iteration 9/10 ----------
1. Generating candidate responses (temp=0.8)...
2. Evaluating candidates...
  Candidate 1: Reward=0.5887, Final Score=0.5931
    Text: Think of entropy as the degree of randomnes
s or unpredictability in a system. Consider a perfect
ly o...
  Candidate 2: Reward=0.5859, Final Score=0.5903
    Text: Think of entropy as the degree of predictab
ility or order in a system. Consider a computer progr
am t...
  Candidate 3: Reward=0.6760, Final Score=0.6804
    Text: Think of entropy as the degree of randomnes
s or unpredictability in a system. Consider a system
of b...
  Candidate 4: Reward=0.6400, Final Score=0.6444
    Text: Think of entropy as the degree of randomnes
s or unpredictability in a system. Consider a perfect
ly b...

3. Updating Policy...
  Selected best candidate: #3
  Replacing worst example in policy (index 1).
--- Iteration Summary | Avg Reward: 0.6226 | Diversit
y: 0.1172 ---
DE-GRPO Iterations:  90%|█| 9/10 [08:13<01:03, 63.86s
---------- DE-GRPO Iteration 10/10 ----------
1. Generating candidate responses (temp=0.8)...
█
```

Fig. 13 Final DE-GRPO run summary—'DE-GRPO Iterations: 100% — 10/10,' final Avg Reward = 0.6245 , Diversity = 0.1836, and saved artifact 'results/entropy_results_1752552191.json.' The table prints reference responses for SFT and GRPO variants for qualitative comparison.

Step 3: Applying the Bound. Since $E_{\pi_k}[R] \leq \omega$ for all k,

$$E_\pi[R] = \sum_{k=1}^{K} \lambda_k E_{\pi_k}[R] \leq \sum_{k=1}^{K} \lambda_k \omega \tag{16}$$

Step 4: Conclusion. Factoring out ω and using $\sum_{k=1}^{K} \lambda_k = 1$:

$$E_\pi[R] \leq \omega \sum_{k=1}^{K} \lambda_k = \omega \tag{17}$$

Thus, no imitation policy can exceed the best expert.

Possibility: The Discovery–Efficacy Tradeoff

Intuition: To break the imitation ceiling, an agent must explore. Exploration is risky, it may lower short-term performance, but creates the possibility of discovering better strategies. This Principle formalizes the tradeoff.

[Discovery–Efficacy Tradeoff] Let the expanded policy space be

$$\Pi_{\text{explore}}(\varepsilon) = (1-\varepsilon)\Pi_{\text{imitate}} + \varepsilon \mathscr{D}_{\text{novel}} \tag{18}$$

where Π_{imitate} is the imitation policy space, $\mathscr{D}_{\text{novel}}$ is a distribution over novel policies, and $\varepsilon \in [0,1]$ is the discovery rate. Let η^* be the maximum achievable efficacy. Then η^* is a monotone non-decreasing function of ε.

Step 1: Boundary Condition. If $\varepsilon = 0$, then $\Pi_{\text{explore}}(0) = \Pi_{\text{imitate}}$. By Principle 1, $\eta^*(0) \leq \omega$.

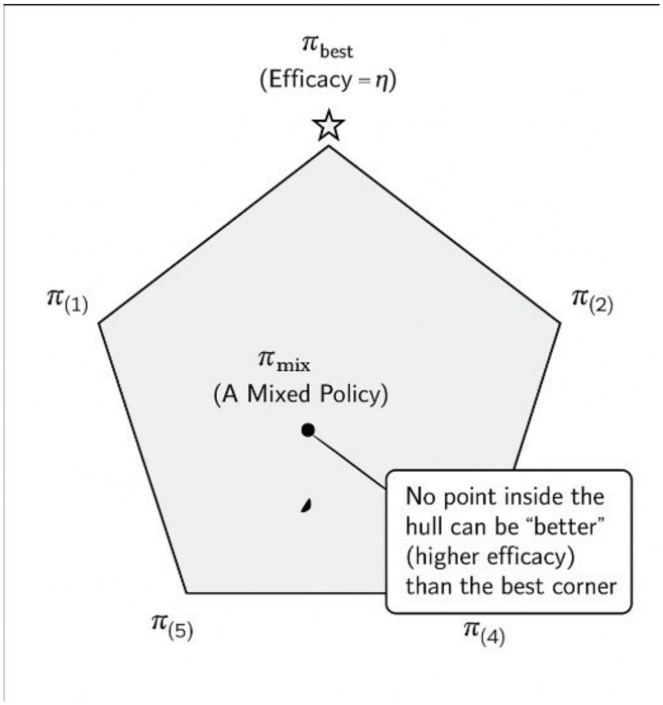


Fig. 14 The Imitation Efficacy Ceiling. Mixing: expert policies cannot create a better policy than the best corner.

Step 2: Monotonicity. For $\varepsilon > 0$, we have $\Pi_{\text{explore}}(\varepsilon) \supset \Pi_{\text{imitate}}$. Thus,

$$\eta^*(\varepsilon) = \max_{\pi \in \Pi_{\text{explore}}(\varepsilon)} E_\pi[R] \geq \max_{\pi \in \Pi_{\text{imitate}}} E_\pi[R] = \omega \tag{19}$$

Hence, $\eta^*(\varepsilon)$ is non-decreasing.

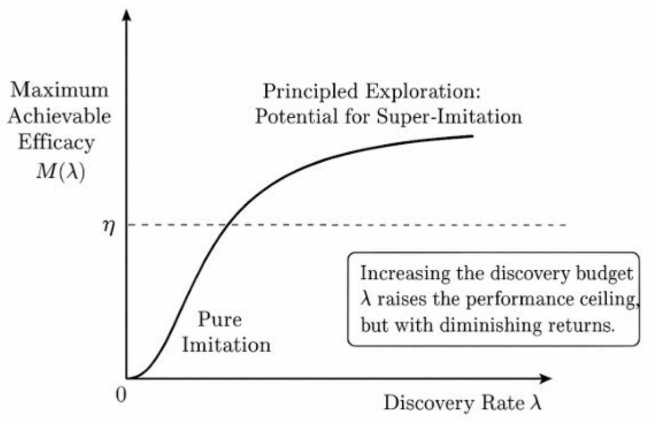


Fig. 15 The Discovery-Efficacy Tradeoff

Mechanism: The Critical Diversity Threshold

Intuition: Exploration requires diversity. Below a critical level, the agent is trapped; above it, breakthroughs become possible.

[Critical Diversity Threshold] Let a population of Teacher policies have diversity D_t at time t. There exists a threshold D_{crit} such that:

1. If $D_t < D_{\text{crit}}$ for all t, the system converges to imitation equilibrium with performance $\eta_\infty \leq \omega$.
2. If $D_t > D_{\text{crit}}$ at some time t^*, the system has positive probability of achieving $\eta > \omega$.

[Proof (Sketch)]

- Define the *imitation basin* as policies with $E_\pi[R] \leq \omega$.
- If $D_t < D_{\text{crit}}$, the chance of generating π with $E_\pi[R] > \omega$ is zero → trapped basin.
- If $D_t > D_{\text{crit}}$, the generative process produces such policies with non-zero probability, and selection spreads them.

Thus, diversity enables escape.

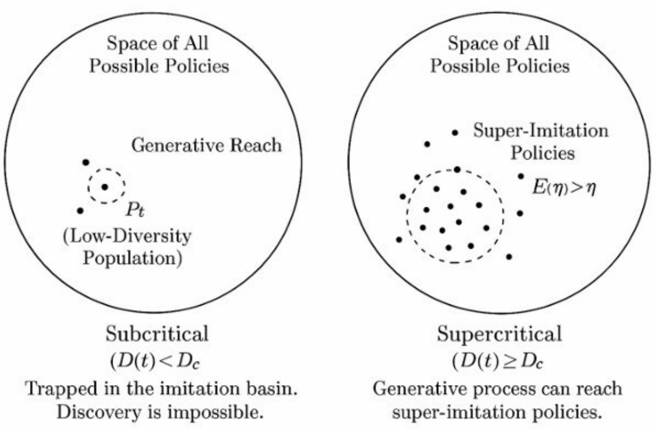

Space of All Possible Policies
Generative Reach
P_t
(Low-Diversity Population)
Subcritical
$(D(t) < D_c$
Trapped in the imitation basin. Discovery is impossible.
Space of All Possible Policies
Super-Imitation Policies
$E(\eta) > \eta$
Supercritical
$(D(t) \geq D_c$
Generative process can reach super-imitation policies.

Fig. 16 Discovery as Phase Transition

Robustness: The PAC-Verifier Guarantee

Intuition: Novelty alone isn't enough, strategies must be validated. A verifier provides robustness by ensuring learned strategies approximate the true reward.

[PAC-Verifier Guarantee] Suppose a verifier provides a noisy reward signal $\tilde{R}$ that is uniformly close to the true reward signal R, such that the error is bounded by ε for all states s and actions a:

$$|\tilde{R}(s,a) - R(s,a)| \leq \varepsilon \tag{20}$$

Let π^* be the optimal policy under the true reward R, and let $\hat{\pi}$ be the policy learned by an agent optimizing the noisy reward $\tilde{R}$. Then the performance gap in terms of true efficacy is bounded:

$$|V^{\pi^*}(s_0) - V^{\hat{\pi}}(s_0)| \leq \frac{2\varepsilon}{1-\gamma} \tag{21}$$

where γ is the discount factor.

[Proof (Sketch)]

- The error in rewards propagates into value functions with bound $\frac{\varepsilon}{1-\gamma}$.
- The performance gap is at most twice this.

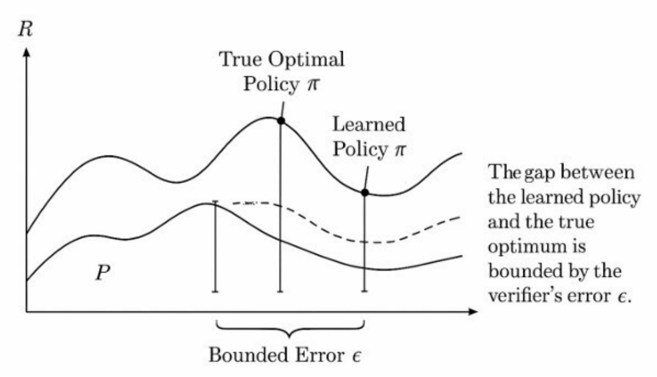

R
True Optimal Policy π
Learned Policy π
The gap between the learned policy and the true optimum is bounded by the verifier's error ϵ.
P
Bounded Error ϵ

Fig. 17 The PAC-Verifier Guarantee

Principle 5 (The 'No Free Lunch' Principle for Pedagogy)

Intuition: This principle synthesizes the previous Principles to argue that reliably breaking the imitation ceiling within our framework requires both a mechanism for creativity (exploration beyond experts) and a mechanism for grounding (a reliable verifier). Our framework suggests that without both of these components, an agent's attempts at discovery are unlikely to be effective or reliable.

[No Free Lunch for Pedagogy] An algorithm $\mathscr{A}$ can guarantee finding a policy π with $E_\pi[R] > \omega$ only if it has:

1. A mechanism to generate policies outside Π_{imitate}.
2. A reward signal correlated with true efficacy.

[Proof (by Contradiction)]

- *Case 1:* If no expansion mechanism, search is confined to Π_{imitate} → by Principle 1, $E_\pi[R] \leq \omega$. Contradiction.
- *Case 2:* If no grounded verifier, reward is uncorrelated with efficacy → by Principle 4, performance gap may be unbounded. Contradiction.

Thus, both are necessary.

Scalability: The Capacity Monotonicity Lemma

Intuition: This lemma is basically about scale. The question we're asking is simple: if we use a bigger, more powerful AI model, can we expect it to actually do better at discovering good teaching strategies? The answer is yes. A model with more capacity (meaning it can represent and try out a wider variety of strategies) will never do worse than a smaller one—and in fact, it has the potential to do better. In other words: the more capable the tool, the higher the ceiling for what it can achieve.

[Capacity Monotonicity] Let two pedagogical agent models, M and M', have corresponding policy spaces Π and Π'. We say model M' has a capacity greater than or equal to model M (denoted $M' \succeq M$) if its policy space is a superset of the other, i.e., $\Pi \subseteq \Pi'$. Let $\eta^*(\Pi)$ be the optimal achievable efficacy over a given policy space Π. If $M' \succeq M$, then their optimal achievable efficacies are ordered accordingly:

$$\eta^*(\Pi') \geq \eta^*(\Pi) \tag{22}$$

Step 1: Define the Optimum for the Less Capable Model

Let $\pi^* \in \Pi$ be an optimal policy for the less capable model, M. By definition, this policy is an element of the policy space Π, and it achieves the maximum possible efficacy within that space:

$$\eta^*(\Pi) = \max_{\pi \in \Pi} E_\pi[R] = E_{\pi^*}[R] \tag{23}$$

Step 2: Relate the Policy Spaces

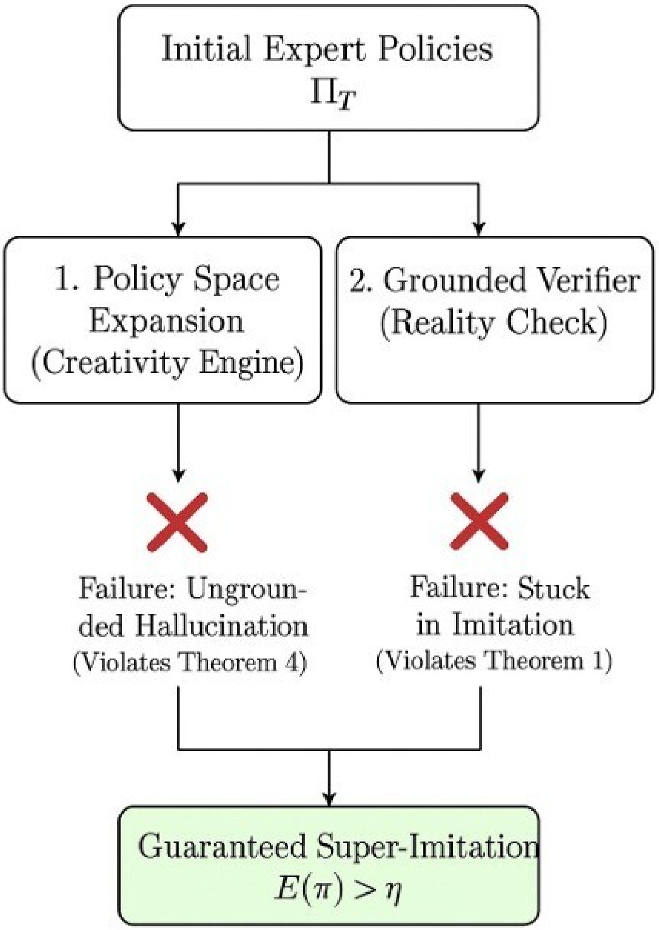

Initial Expert Policies
Π_T
1. Policy Space Expansion (Creativity Engine)
2. Grounded Verifier (Reality Check)
Failure: Ungrounded Hallucination
(Violates Theorem 4)
Failure: Stuck in Imitation
(Violates Theorem 1)
Guaranteed Super-Imitation
$E(\pi) > \eta$

Fig. 18 The Necessary Path to Super-Imitation

By the definition of model capacity, since $M' \succeq M$, we have the set inclusion $\Pi \subseteq \Pi'$. This means that any policy that can be expressed by model M can also be expressed by model M'. Therefore, the optimal policy for M, π^*, must also be an element of the policy space for M':

$$\pi^* \in \Pi' \tag{24}$$

Step 3: Apply the Definition of the Maximum

The optimal achievable efficacy for the more capable model, M', is $\eta^*(\Pi')$, which is the maximum efficacy over *all* policies in its space Π'. Since π^* is one such policy in that space, the maximum efficacy for M' must be greater than or equal to the efficacy of this particular policy:

$$\eta^*(\Pi') = \max_{\pi \in \Pi'} E_\pi[R] \geq E_{\pi^*}[R] \tag{25}$$

Step 4: Conclusion

From Step 1, we know that $E_{\pi^*}[R] = \eta^*(\Pi)$. Substituting this into the inequality from Step 3, we arrive at the final result:

$$\eta^*(\Pi') \geq \eta^*(\Pi) \tag{26}$$

This proves that the optimal achievable efficacy is a monotone non-decreasing function of model capacity. A more capable model, by virtue of having access to all the strategies of a less capable model plus potentially more, cannot yield a worse optimal outcome.

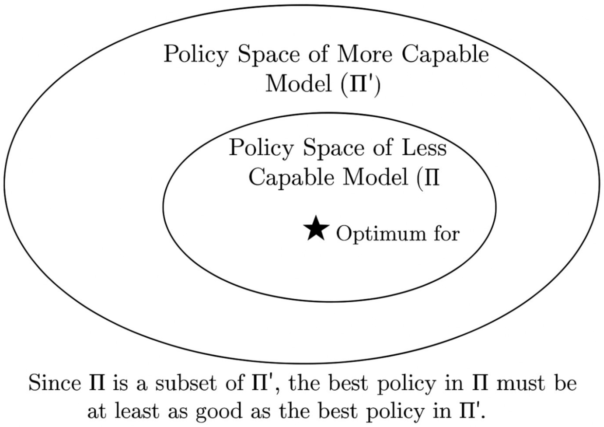

Policy Space of More Capable Model (Π')
Policy Space of Less Capable Model (Π
★ Optimum for
Since Π is a subset of Π', the best policy in Π must be at least as good as the best policy in Π'.

Fig. 19 Policy space inclusion, π' (less capable) is a subset of π' (more capable), ensuring π' can achieve equal or better optimal performance.