1. Problem & Motivation

In complex decision-making scenarios - such as strategic planning, debugging code, or writing academic arguments - humans often suffer from **cognitive fixation**: the tendency to get stuck on a single line of thought or the first available solution. While Large Language Models (LLMs) like ChatGPT have become powerful assistants, they paradoxically reinforce this issue. Most current interfaces rely on a linear "Chat" paradigm, which encourages **Chain of Thought (CoT)** reasoning. This linear approach provides a single, probable answer, often hiding alternative possibilities and leading to **homogenization** of ideas (Anderson et al., 2024).

The problem we are tackling is the **lack of agency and exploration** in current Human-AI interactions. Users are often passive consumers of a generated result, rather than active partners in a thought process. There is a need for a system that helps users explore divergent paths, and make informed choices, rather than just receiving a final output.

2. System Objectives

Our system, **Lantern**, is designed to shift Human-AI interaction from a passive, linear dialogue to an active, collaborative exploration. Our core objectives are:

- **Facilitate Divergent Thinking:** To break cognitive fixation, the system generates multiple, distinct options at every step rather than a single output, encouraging the exploration of diverse directions.
- **Empower User Agency & Alignment:** The user actively navigates the reasoning process by "pruning" or expanding branches. This selection process establishes a shared mental model, ensuring the AI remains synchronized with the user's specific intent.
- **Mitigate Automation Bias & Sycophancy:** To prevent over-reliance on agreeable AI responses, a "Devil's Advocate" mechanism challenges the user's chosen path. This fosters critical reflection and deliberate decision-making over passive acceptance.

3. Existing Technology and Theoretical Review

Existing Tools: The landscape of AI writing support is currently dominated by two paradigms, neither of which supports deep cognitive exploration:

- **Corrective & Optimization Tools (e.g., Grammarly, Wordtune):** These tools excel at **refinement**. They optimize existing text for grammar, tone, and clarity, aiming to bring the user's input closer to a standard or "correct" form. However, they operate on the surface level (syntax/style) and do not challenge the user's underlying logic, narrative choices, or biases.
- **Generative "Ghostwriters" (e.g., Jasper, Sudowrite, ChatGPT):** These tools focus on automation. They generate entire paragraphs or plot points to "unblock" the user. While efficient, they often reduce user agency and lead to homogenization, as the user becomes an editor of AI-generated content rather than the primary thinker.

Lantern innovation lies in behaviorally-trained LLM agents, guided through explicit prompt-level behavioral conditioning(rather than fine-tuning). This creates a system whose focus is not writing for the user, but shaping the user's reasoning process through structured divergence, deliberate critical opposition, and behaviorally-conditioned agents trained on widely accepted principles of academic writing (clarity, coherence, argumentative structure, and evidence-based reasoning).

Theoretical Grounding

Lantern is grounded in research on the cognitive and structural limitations of LLMs and their influence on human reasoning. LLMs operate as autoregressive next-token predictors (Brown et al., 2020), which produces *maximization bias*: a tendency to generate highly probable, generic continuations that narrow creative exploration (Holtzman et al., 2019). In addition, conversational LLMs often display sycophancy-aligning with or reinforcing the user's beliefs, sometimes even contradicting their own statements in order to appear helpful (Sponheim, 2024).

To counter these tendencies, Lantern draws on the Tree-of-Thoughts framework (Yao et al., 2023), enabling exploration of multiple reasoning paths rather than a single linear CoT trajectory. From Creativity Support Tools (CST), we incorporate principles of divergent thinking by generating conceptually distinct alternatives instead of converging on one dominant output. Finally, a Devil's Advocate agent, grounded in Dual Process Theory (Chiang et al., 2024), challenges the user's chosen branch to mitigate automation bias and reduce sycophancy-driven agreement-shifting users from fast, intuitive System-1 acceptance toward slow, deliberate System-2 evaluation.

4. Approach (Intelligence Design)

At the intelligence level, Lantern orchestrates multiple LLM-based agents using behavioral prompt engineering rather than fine-tuning. We implement distinct functional roles, such as Divergent Generators that expand the solution space and Devil's Advocate agents that critique user biases. This design fosters meaningful human interaction by establishing a dialectic relationship: the system resists sycophantic agreement, forcing the user to actively evaluate conflicting alternatives rather than passively accepting output. By encoding academic standards into these opposing prompts, the intelligence layer transforms the writing process into a critical co-creation loop, ensuring the user retains cognitive autonomy while engaging with the AI as a challenging partner.

To keep the interaction lightweight, these agents operate in an iterative cycle: the system presents a small set of contrasting reasoning paths, the user selects or rejects them, and the agents regenerate refined alternatives in response.

5. Plan:

Dates	Tasks & Milestones
Dec 9 - Dec 23	• research the rules of academic writing and criticing • Develop basic backend logic for Gen-Eval loop (generation & scoring). • Set up Streamlit environment and basic UI layout. • Milestone 2 Submission - Initial Mockup.
Dec 24 - Jan 10	• Implement the visualization. • Connect UI to live API. • Implement interactive "Expand Node" functionality.
Jan 11 - Jan 20	• Refine prompts to ensure diverse outputs (Prompt Engineering).
Jan 21 - Jan 29	• Conduct user evaluation (5 users) testing decision-making scenarios. • Final debugging and report writing. • Final Submission.