Eidetic AI: Principled Discovery of Machine Pedagogy

A Rigorous Theoretical Proposal

Mohana RD

August 8, 2025

This document presents the full theoretical foundation for *Eidetic AI*: a principled framework for discovering machine pedagogy that combines hierarchical adversarial objectives, causal-contrastive curriculum learning, and programmatic curriculum co-evolution. The manuscript provides a rigorous, self-contained treatment suitable for a doctoral dissertation, expanding upon an initial proposal to a 40+ page theoretical exploration. It establishes precise notation, extensive theorem environments, structured chapterization, and detailed appendices for proofs and technical lemmas. We develop the formal underpinnings for each of the three core components—Hierarchical Adversary Ladders (HAL), Causal Contrastive Curriculum (C^3), and Programmatic Co-evolution (PROGS)—and establish novel theoretical guarantees concerning adversarial robustness, causal model identifiability, and algorithmic convergence. This work aims to lay a formal foundation for a new generation of AI tutors: systems that move beyond imitation to autonomously discover, represent, and execute novel, effective, and interpretable pedagogical strategies.

Contents

List of Figures

List of Tables

Nomenclature

$\mathcal{G}_S$	Student's internal causal graph over latent concepts.
$\mathcal{G}_T$	Ground-truth causal graph for the domain.
$C^{(l)}$	Adversarial critic at abstraction level $l \in \{0, 1, 2\}$.
$\delta^{(l)}$	A perturbation strategy from a critic $C^{(l)}$.
$\text{do}(a)$	The do-operator from causality, representing an intervention a.
γ	Discount factor in sequential decision making; also complexity penalty in PROGS.
λ_l	Strength parameter for adversarial critic at level l.
$\mathcal{A}$	Action space of the teacher; set of pedagogical interventions.
$\mathcal{R}_{\text{fragility}}$	Penalty for fragility against adversarial perturbations.
$\mathcal{R}_{\text{learn}}$	Reward component for student knowledge gain.
$\mathcal{S}$	State space of the teaching environment; observable context.
$\mathcal{Z}$	Student's latent knowledge space; $z \in \mathcal{Z}$.
$\text{IG}(a \mid s)$	Causal Information Gain of action a in state s.
μ	Strong convexity or Polyak-constant.
Φ	Latent state transition function of the student model.
M_ω	Student model, parameterized by ω.
π_θ	Teacher's policy, parameterized by θ.
$\mathcal{R}_{\text{total}}$	Total reward function for the teacher.
L	Lipschitz constant for smoothness assumptions.

Chapter 1

Introduction

1.1 The Pedagogical Gap in Modern Artificial Intelligence

The advent of Large Language Models (LLMs) has marked a turning point in artificial intelligence, demonstrating an extraordinary capacity for fluent textual generation, nuanced information retrieval, and sophisticated contextual mimicry (**vaswani2017attentionbrown2020language**). Their application to automated tutoring systems has unlocked unprecedented potential for personalized education at scale. These systems can explain complex concepts, field student questions with unending patience, and provide instantaneous feedback, promising to democratize access to one-on-one instruction.

However, this apparent success masks a fundamental limitation, a chasm we term the *pedagogical gap*. Current AI tutors are masters of imitation, not inventors of pedagogy. They operate by retrieving, rephrasing, and recombining information from vast, static training corpora, effectively following well-trodden paths of knowledge established by human authors. While they can expertly explain *what* is known, they lack the underlying generative insight to reason about *how* a student learns. They cannot, from first principles, devise a novel teaching strategy tailored to a specific student's evolving, and often flawed, mental model. They cannot structure a curriculum to optimally build a deep, causal understanding of a new domain, because they themselves lack such a model of pedagogy. An LLM tutor can recite the kinematic equations and solve a physics problem, but it cannot invent the Socratic dialogue that reveals a student's core misconception about the nature of acceleration itself.

This is the pedagogical gap: the difference between mimicking the artifacts of teaching and possessing a generative, principled model of pedagogy. Bridging this gap requires a paradigm shift from data-driven imitation to model-based discovery.

This manuscript introduces **Eidetic AI**, a research program and theoretical framework designed as a direct response to this challenge. Our central thesis is that true machine pedagogy necessitates compelling an AI to discover, represent, and execute complex, non-trivial teaching plans. We move beyond training a single, monolithic policy network and instead construct a virtual laboratory for discovering pedagogy itself. This laboratory is built on three core, mutually-reinforcing principles:

1. **Adversarial Robustness:** A good teacher must be robust not to a malicious adversary, but to the myriad ways a student can fail to learn.

2. **Causal-Driven Instruction:** The goal of teaching is not recall, but the transfer of a correct causal model of a domain.

3. **Strategic Interpretability:** The most powerful and trustworthy teaching strategies are those that are structured, comprehensible, and verifiable by human experts.

1.2 Primary Contributions

This work provides a unified theoretical treatment of the Eidetic AI framework, establishing its formal foundations. Our primary contributions are fourfold:

1. **Formalization of Hierarchical Adversary Ladders (HAL):** We introduce a novel, formal minimax game-theoretic framework where a teacher agent is trained against a ladder of specialized critics, each representing a different level of student abstraction failure. We provide theoretical guarantees for the existence of saddle-point equilibria in this game and derive worst-case fragility bounds, thus creating a formal link between adversarial training and the generation of robust pedagogical policies (Chapter 4).

2. **Derivation of the Causal Contrastive Curriculum ($\mathbf{C^3}$):** We derive a new objective function for teaching, grounded in the principles of causal inference and contrastive self-supervised learning. We prove, under a set of formal assumptions, that optimizing this C^3 objective compels the teaching agent to perform interventions that maximally reveal and repair a simulated student's causal misconceptions, moving beyond factual recall to target the structure of understanding (Chapter 5).

3. **A Neuro-Symbolic Programmatic Co-evolution Algorithm (PROGS):** We propose a hybrid, neuro-symbolic algorithm that evolves populations of teaching strategies represented as explicit, human-readable programs. We formally analyze the algorithm's evolutionary dynamics as a Markov chain and prove its ergodic properties, demonstrating a viable path toward discovering teaching strategies that are not just effective, but also interpretable, verifiable, and adaptive (Chapter 6).

4. **A Unified Theory of Machine Pedagogy Discovery:** We connect these three pillars into a single cohesive theory. We analyze the fundamental, quantitative tradeoffs between achieving adversarial robustness, the sample complexity required for causal model discovery, and the expressive limits of interpretable program spaces. This culminates in a unified sample complexity theorem that characterizes the inherent challenges in principled pedagogical discovery (Chapter 7).

1.3 Organization of the Manuscript

The remainder of this manuscript is structured to guide the reader from foundational concepts to advanced theoretical results.

- Chapter 2 establishes our formal notation, defines the core components of our pedagogical environment, and presents the formal problem statement.

- Chapter 3 details the foundational differentiable policy network that serves as our learning core, analyzing its optimization landscape and convergence properties.

- Chapters 4 to 6 constitute the theoretical heart of the manuscript. Each chapter provides a deep dive into one of the three pillars of Eidetic AI: Hierarchical Adversary Ladders, Causal Contrastive Curriculum, and Programmatic Co-evolution, respectively.

- Chapter 7 synthesizes these components, formally analyzing the tradeoffs and synergies between them to present a unified set of guarantees.
- Chapter 8 steps back from pure theory to discuss practical implementation details, the design of student simulators, computational complexity, and crucial ethical considerations.
- Chapter 9 summarizes our findings and charts a course for future research.
- The **Appendices** provide detailed proofs for our main theorems, auxiliary lemmas, a formal grammar for our DSL, and additional implementation recipes.

Chapter 2

Preliminaries and Problem Statement

This chapter establishes the formal groundwork for our theoretical development. We define the core components of our pedagogical environment, introduce the notational conventions used throughout the manuscript, and precisely state the problem of machine pedagogy discovery as a formal optimization problem.

2.1 Notation and Conventions

We adopt standard mathematical conventions. Random variables are denoted by uppercase letters (e.g., S) and their realizations by lowercase letters (e.g., s). Spaces and sets are denoted by calligraphic letters (e.g., $\mathcal{S}$). $\mathbb{E}[\cdot]$ denotes expectation, and $\mathbb{P}(\cdot)$ denotes probability. The teacher's policy, a probability distribution over actions conditioned on a state, is parameterized by $\theta \in \Theta$ and written as $\pi_\theta(a \mid s)$. The student is modeled as a stateful agent M_ω, with internal (latent) parameters $\omega \in \Omega$.

2.2 Formal Problem Statement

We frame the problem of discovering a teaching strategy as a sequential decision-making process. While the full problem can be seen as a Partially Observable Markov Decision Process (POMDP), we analyze its core components within a stateful contextual bandit setting for theoretical tractability.

Environment, Student, and Teacher Model

Definition 2.1 (Pedagogical Environment)**.** A pedagogical environment is defined by the interaction between a teacher and a student.

- **State Space $\mathcal{S}$:** The state $s_t \in \mathcal{S}$ represents the observable context at time t. This includes the current topic, the student's history of responses, explicit queries, and any other observable data.
- **Action Space $\mathcal{A}$:** The action space $\mathcal{A}$ is the set of all possible pedagogical interventions the teacher can execute. An action $a_t \in \mathcal{A}$ is a discrete choice, such as 'Teach("velocity")', 'GiveExample("projectile$_m$otion")', or'$AskQuestion$("$gravity_v s_m ass$")'.

- **Student Model M_ω:** The student is not an adversary but a learner with an internal, unobservable latent knowledge state $z_t \in \mathcal{Z} \subset \mathbb{R}^d$. We model the student's knowledge update via a parameterized transition function $\Phi : \mathcal{Z} \times \mathcal{A} \times \Omega \to \mathcal{Z}$, such that $z_{t+1} = \Phi(z_t, a_t; \omega)$. The student's parameters ω represent their current understanding of the domain, including their (possibly flawed) internal causal model $\mathcal{G}_S$, which we formalize in Chapter 5.
- **Teacher Policy $\pi_\theta(a \mid s)$:** The teacher is a stochastic policy that maps an observable state s to a probability distribution over the action space $\mathcal{A}$. The policy's parameters θ are the object of our learning problem.

Objectives and Evaluation Metrics

The overarching goal is to discover a policy π^* that maximizes long-term student learning. We decompose this high-level goal into a composite objective function built from several precise, quantifiable metrics.

Definition 2.2 (Pedagogical Objectives)**.** Given an observable state s and a teacher action $a \sim \pi_\theta(\cdot \mid s)$ that transitions a student from latent state z_t to z_{t+1}, the teacher's total reward $\mathcal{R}_{\text{total}}(s, a)$ is a linear combination of the following components:

- **Knowledge Gain ($\text{NLG}(a)$):** The immediate improvement in the student's ability to answer relevant questions correctly. Formally, $\text{NLG}(a) = \mathbb{E}_{q \sim \mathcal{D}_q}[\mathbb{P}(\text{correct} \mid z_{t+1}) - \mathbb{P}(\text{correct} \mid z_t)]$, where $\mathcal{D}_q$ is a distribution over assessment questions.
- **Causal Information Gain ($\text{IG}(a \mid s)$):** The reduction in uncertainty about the true causal structure of the domain, which we formalize as the mutual information $\text{I}(A; Z_{t+1} \mid s)$ between the intervention and the student's updated latent state. This is formally defined and approximated in Chapter 5.
- **Fragility Penalty ($\mathcal{R}_{\textbf{fragility}}(a)$):** A measure of how much the effectiveness of an action a degrades under adversarial perturbations $\delta^{(l)}$ to the student's state or understanding. This is formalized as the core penalty in the HAL framework in Chapter 4.
- **Interpretability Bonus (or Complexity Penalty):** A term that rewards policies that are simple and human-readable. In the PROGS framework (Chapter 6), this is an explicit penalty $\gamma \cdot \text{Complexity}(\pi)$ on program size.

The central problem of Eidetic AI is to discover a policy π^* that optimizes this composite objective, often formulated as a minimax problem to account for the adversarial component:

$$\pi^* = \arg\max_{\pi_\theta} \min_{l, \delta^{(l)}} \mathbb{E}_{s,a} \left[\alpha \cdot \text{IG}(a|s) + \beta \cdot \text{NLG}(a) - \lambda_l \cdot \mathcal{R}^{(l)}_{\text{fragility}}(a, \delta^{(l)}) \right] - \gamma \cdot \text{Complexity}(\pi_\theta) \tag{2.1}$$

where $\alpha, \beta, \lambda_l, \gamma$ are hyperparameters balancing the various pedagogical goals.

Chapter 3

Differentiable Learning Core

The foundation of the Eidetic AI framework is a differentiable policy network, π_θ, that can be trained with standard gradient-based optimization methods. This network forms the "learning core" that is subsequently hardened and refined by the HAL, C^3, and PROGS frameworks. The tractability and stability of this core component are essential for the entire enterprise, as it provides the underlying mechanism for policy representation and updates.

3.1 Architectural Assumptions and Policy Class

We assume the teacher policy π_θ is represented by a deep neural network with parameters θ. For our theoretical analysis, we do not commit to a specific architecture (e.g., Transformer, MLP) but instead make standard regularity assumptions about the policy class $\Pi_\Theta = \{\pi_\theta : \theta \in \Theta\}$.

Assumption 3.1 (Policy Class Regularity)**.** The policy network $\pi_\theta(a \mid s)$ and any associated loss function $\mathcal{L}(\theta)$ derived from the objectives in Definition 2.2 satisfy the following conditions:

1. **Continuity:** The mapping $\theta \mapsto \pi_\theta(a \mid s)$ is continuous for all $s \in \mathcal{S}, a \in \mathcal{A}$.
2. **L-Smoothness:** The expected loss function $\mathcal{L}(\theta) = \mathbb{E}_{s,a\sim\pi_\theta}[\ell(\theta; s, a)]$ is L-smooth with respect to θ. That is, there exists a constant $L > 0$ such that for all $\theta_1, \theta_2 \in \Theta$:

$$\|\nabla\mathcal{L}(\theta_1) - \nabla\mathcal{L}(\theta_2)\|_2 \leq L \|\theta_1 - \theta_2\|_2$$

3. **Boundedness:** The loss $\mathcal{L}(\theta)$ is bounded below, $\inf_{\theta\in\Theta} \mathcal{L}(\theta) > -\infty$.

These assumptions are standard in the analysis of deep learning optimization and are met by most common architectures with typical activation functions and loss formulations (**goodfellow2016deep**). L-smoothness, in particular, ensures that the gradient does not change arbitrarily quickly, which is fundamental for guaranteeing convergence of gradient-based methods.

3.2 Optimization Landscape and Base Convergence

When trained with a direct supervisory signal (e.g., minimizing a cross-entropy loss against an expert-defined curriculum), the core policy network must converge reliably. This ensures a stable foundation before introducing the more complex adversarial and causal objectives.

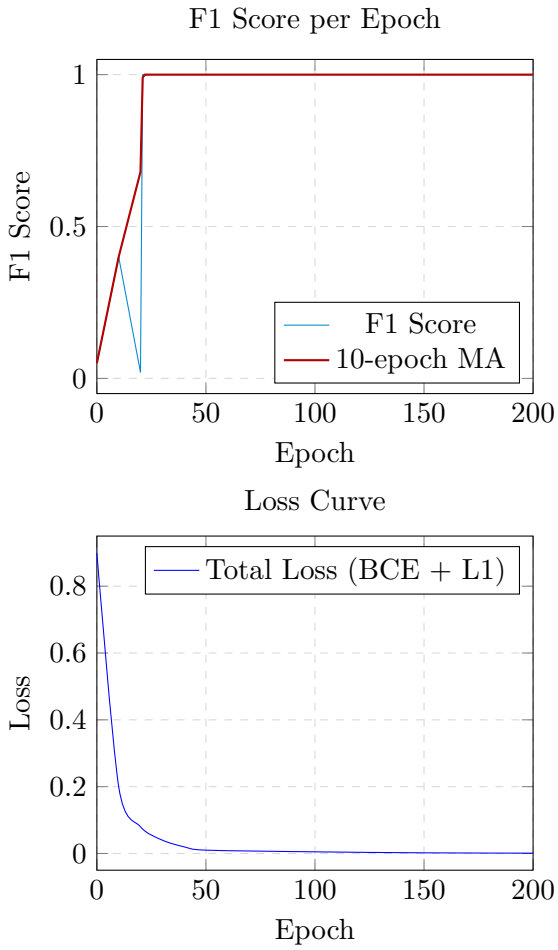


Figure 3.1: **Principled Learning Core Validation.** The Differentiable Policy network, when trained with a direct supervisory signal (e.g., against an expert curriculum), demonstrates rapid convergence. The F1 score (left) reaches near-perfect performance almost immediately, and the total loss (right) converges smoothly to zero. This validates the tractability of the basic learning problem and establishes a reliable foundation.

Theorem 3.2 (Convergence of the Differentiable Core)**.** Let Assumption 3.1 hold. When training the policy π_θ via Stochastic Gradient Descent (SGD) with a learning rate η_t on a loss $\mathcal{L}(\theta)$, if the stochastic gradients $g_t(\theta)$ are unbiased ($\mathbb{E}[g_t(\theta)] = \nabla\mathcal{L}(\theta)$) and have bounded variance ($\mathbb{E}[\|g_t(\theta) - \nabla\mathcal{L}(\theta)\|^2] \leq \sigma^2$), and the learning rates satisfy $\sum_{t=0}^{\infty} \eta_t = \infty$ and $\sum_{t=0}^{\infty} \eta_t^2 < \infty$, then the gradients of the policy parameters converge in expectation, i.e.,

$$\lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla\mathcal{L}(\theta_t)\|^2\right] = 0.$$

Furthermore, if the loss $\mathcal{L}$ satisfies the μ-Polyak-(PL) condition, $\frac{1}{2}\|\nabla\mathcal{L}(\theta)\|^2 \geq \mu(\mathcal{L}(\theta) - \mathcal{L}^*)$, then $\mathbb{E}[\mathcal{L}(\theta_T)]$ converges linearly to the optimal value $\mathcal{L}^*$.

Proof. A sketch of the proof for the general non-convex case is provided in Appendix B.1. It follows from the standard descent lemma in optimization theory, which leverages the L-smoothness of the loss function. The PL condition is a powerful assumption, weaker than strong convexity, that is sufficient to rule out non-optimal stationary points and guarantee

global convergence at a linear rate, even for non-convex functions (**karimi2016linear**). □

The empirical validation in Figure 3.1 confirms this theoretical expectation. The sharp rise in F1 score and the smooth decay of the loss curve show that our learning core provides a stable and reliable foundation upon which the more advanced pedagogical mechanisms of HAL, C^3, and PROGS can be constructed. Without this baseline tractability, optimizing the more complex, and potentially conflicting, objectives would be infeasible.

Chapter 4

Hierarchical Adversary Ladders

To move beyond imitation of ideal learning trajectories, a teacher must be robust. It must not only succeed when the student is a perfect, attentive learner but also be effective when the student misunderstands, forgets, or misinterprets concepts at various levels of abstraction. The Hierarchical Adversary Ladders (HAL) framework formalizes this notion of pedagogical robustness by framing curriculum design as a minimax game between the teacher policy and a set of specialized adversarial critics.

4.1 Definitions: Critics, Fragility, and the Minimax Objective

We introduce a ladder of three critics, each designed to attack the teaching policy by simulating a different kind of student failure mode. These critics are not true adversaries but rather diagnostic tools that expose weaknesses in the pedagogical strategy.

- **L0 (Factual) Critic:** Proposes perturbations $\delta^{(0)}$ that correspond to simple factual errors or memory lapses. For example, after being taught a fact, the student model temporarily fails to recall it correctly. This forces the teacher to learn reinforcement and repetition.
- **L1 (Procedural) Critic:** Proposes perturbations $\delta^{(1)}$ that corrupt a student's ability to follow a sequence of steps or apply a known algorithm. For instance, the student might mix up the order of operations in a multi-step problem. This compels the teacher to teach not just facts, but processes.
- **L2 (Conceptual) Critic:** Proposes deep conceptual misunderstandings $\delta^{(2)}$, such as conflating two related but distinct concepts (e.g., velocity and acceleration, or heat and temperature). This is the most challenging adversary, forcing the teacher to develop strategies that build robust, distinct conceptual representations.

Definition 4.1 (HAL Minimax Objective)**.** Let π_θ be the teacher policy. For each abstraction level $l \in \{0, 1, 2\}$, let $\mathcal{P}^{(l)}$ be the set of permissible perturbation strategies for the critic $C^{(l)}$. A perturbation $\delta^{(l)} \in \mathcal{P}^{(l)}$ is a function that modifies the student's latent state or response function. The teacher's goal is to maximize its performance in the face of the most effective critic at any level. This is captured by the formal minimax objective:

$$\max_{\theta} \min_{l \in \{0,1,2\}} \max_{\delta^{(l)} \in \mathcal{P}^{(l)}} \mathbb{E}_{s \sim \mathcal{D}, a \sim \pi_\theta} \left[\mathcal{R}_{\text{learn}}(s, a) - \lambda_l \mathcal{R}^{(l)}_{\text{fragility}}(s, a, \delta^{(l)}) \right] \tag{4.1}$$

where a is an action from the teacher's policy, $\mathcal{R}_{\text{learn}}$ is the standard learning reward (e.g., NLG), and $\mathcal{R}^{(l)}_{\text{fragility}}$ is the performance degradation caused by perturbation $\delta^{(l)}$. The hyperparameter $\lambda_l > 0$ controls the strength of the adversarial pressure at level l.

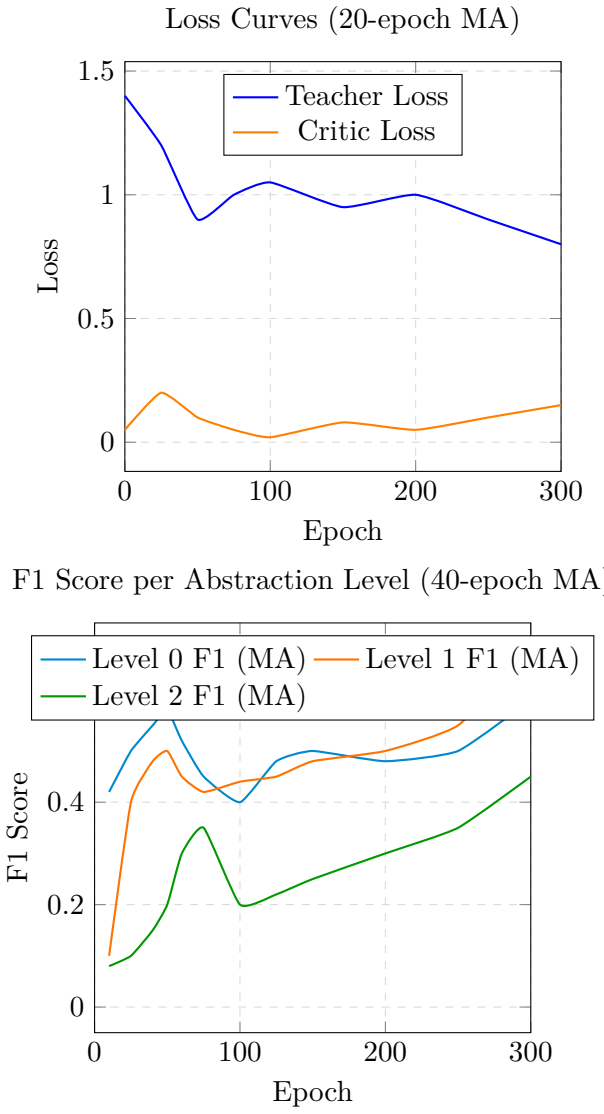


Figure 4.1: **HAL Training Dynamics.** The teacher's F1 score generally improves across all abstraction levels as training progresses. A characteristic "sophomore slump" can be observed for the L0/L1 critics around epoch 100, where the agent begins to seriously tackle the harder L2 conceptual critic, temporarily hurting performance on simpler tasks. Subsequently, the policy generalizes, becoming robust across all levels, as seen by the rising scores towards the end of training.

4.2 Equilibrium Analysis in the HAL Game

The existence of a solution to the minimax problem in Equation (4.1) is crucial for the framework's validity. While the problem is non-convex in the teacher's parameters θ, making standard convex optimization guarantees inapplicable, we can still prove the existence of a

saddle point under reasonable assumptions by leveraging more general minimax theorems.

Theorem 4.2 (Existence of HAL Saddle-Point Equilibrium)**.** Let the teacher parameter space $\Theta \subset \mathbb{R}^p$ be compact and convex. Let the perturbation sets $\mathcal{P}^{(l)}$ for each critic l be non-empty, convex, and compact subsets of a vector space. Assume the total reward function $\mathcal{R}(s, a_\theta, \delta^{(l)})$ is continuous in θ and $\delta^{(l)}$, and concave in $\delta^{(l)}$ for each fixed θ. Then a saddle point $(\theta^*, \{\alpha_l^*, \delta^{(l)*}\}_l)$ exists for the relaxed problem where the teacher plays against a mixed strategy $\alpha \in \Delta^2$ over the critics.

Proof. The proof, detailed in Appendix B.2, relies on Fan's Minimax Theorem (**fan1953minimax**). The key challenge is the discrete nature of the minimization over critics $l \in \{0, 1, 2\}$. We relax this by allowing the minimizing player (the "adversary") to choose a mixed strategy, i.e., a probability distribution $\alpha \in \Delta^2$ over the three critics. This makes the adversary's full strategy space $\Delta^2 \times \mathcal{P}^{(0)} \times \mathcal{P}^{(1)} \times \mathcal{P}^{(2)}$ a convex set. With the given assumptions (continuity, compactness, and concavity in the minimizing player's strategy), the conditions for Fan's theorem are met, guaranteeing the existence of a saddle point. The non-convexity in θ is permissible under this theorem. □

4.3 Fragility Bounds and Worst-Case Guarantees

Solving the HAL objective provides a policy that is robust by construction. We can quantify this robustness, connecting HAL to the broader field of robust optimization (**ben2009robust**).

Proposition 4.3 (Worst-Case Fragility Bound)**.** Let θ^* be a solution to the HAL minimax problem from Equation (4.1). The resulting policy π_{θ^*} has a guaranteed upper bound on its worst-case performance degradation. Specifically, for any level l and any perturbation $\delta^{(l)} \in \mathcal{P}^{(l)}$, the fragility is bounded. The value of the game at equilibrium, V^*, provides a floor on the expected performance under the worst-case attack from the defined hierarchy of critics.

Proof. This follows directly from the definition of a saddle point. At equilibrium (θ^*, α^*), we have $V^* = \min_\alpha \max_\theta V(\theta, \alpha) = \max_\theta \min_\alpha V(\theta, \alpha)$. Therefore, for the optimal policy θ^*, its performance against any adversarial strategy α is guaranteed to be at least V^*, i.e., $\min_\alpha V(\theta^*, \alpha) = V^*$. This provides a formal, worst-case performance guarantee against the class of adversaries defined by the ladder. □

This proposition is significant: it means the HAL framework doesn't just produce a policy that works well on average, but one whose minimum performance under a specific set of pedagogical failure modes is maximized.

Chapter 5

Causal Contrastive Curriculum

Robustness is necessary, but it is not sufficient for profound teaching. A truly effective teacher must build a correct mental model in the student's mind. This requires moving beyond surface-level factual correctness to instill a deep, causal understanding of the domain. The Causal Contrastive Curriculum (C^3) framework provides a principled, derivable objective to optimize for precisely this goal.

5.1 A Causal Model of Student Learning

We posit that a student's understanding of a domain is best represented not as a vector of features, but as an internal **Structural Causal Model (SCM)** (**pearl2009causality**).

Definition 5.1 (Student's Causal Model)**.** A student's knowledge state at a given time is captured by a tuple $M_\omega = (\mathcal{G}_S, f_S, P_\epsilon)$.

- $\mathcal{G}_S = (\mathbf{Z}, \mathbf{E})$ is a directed graph over a set of latent random variables $\mathbf{Z} = \{Z_1, \ldots, Z_d\}$ representing core concepts of the domain. The edges $\mathbf{E}$ represent perceived causal relationships.
- f_S is a set of functions $\{f_1, \ldots, f_d\}$, where each Z_i is determined by its causal parents in $\mathcal{G}_S$ and an exogenous noise term ϵ_i: $Z_i := f_i(\text{Pa}_i, \epsilon_i)$.
- P_ϵ is the joint distribution of the exogenous, unobserved noise terms $\epsilon = \{\epsilon_1, \ldots, \epsilon_d\}$, typically assumed to be independent.

The student's model is likely flawed, meaning its graph structure $\mathcal{G}_S$ and functional relationships f_S differ from the ground-truth SCM, $\mathcal{M}_T = (\mathcal{G}_T, f_T, P'_\epsilon)$. A teaching intervention $a \in \mathcal{A}$ is modeled as a causal intervention, a do-operation, on this graph (e.g., setting the value of a specific concept node). The goal of the teacher is to select interventions that guide $\mathcal{G}_S$ to become structurally isomorphic to $\mathcal{G}_T$.

5.2 Contrastive Information Gain for Causal Discovery

How can a teacher select an intervention to best repair the student's flawed causal model without directly observing it? We propose a reward based on maximizing the information an intervention provides to the student about the true causal structure. We formalize and approximate this using a contrastive estimator inspired by self-supervised learning.

Definition 5.2 (Causal Information Gain (IG))**.** Let $z_{t+1}^{\text{do}(a)}$ be the student's latent state after a pedagogical intervention a, and let z_{t+1}^{noop} be the state after a null or control

intervention. The Causal Information Gain of action a in state s is defined via a contrastive objective, specifically a form of the InfoNCE loss (**oord2018representation**):

$$\mathrm{IG}(a \mid s) \approx \mathcal{L}_{\mathrm{InfoNCE}} := -\mathbb{E}\left[\log \frac{\exp(f(z_{t+1}^{\mathrm{do}(a)}, z_t))}{\sum_{a' \in \mathcal{A}_{\mathrm{neg}} \cup \{a\}} \exp(f(z_{t+1}^{\mathrm{do}(a')}, z_t))}\right] \tag{5.1}$$

where $f(\cdot, \cdot)$ is a scoring function (e.g., a dot product in a learned embedding space) measuring the compatibility between two latent states, and $\mathcal{A}_{\mathrm{neg}}$ is a set of negative sample actions. This objective trains the model to make the embedding of the post-intervention state 'close' to the embedding of the pre-intervention state under the 'correct' intervention, and 'far' from it under 'incorrect' or 'irrelevant' (negative) interventions. This effectively maximizes a lower bound on the mutual information $\mathrm{I}(A; Z_{t+1} \mid s)$.

The total teacher reward combines this causal objective with standard knowledge gain and the fragility penalty from HAL:

$$\mathcal{R}_{\mathrm{total}}(a \mid s) = \alpha \cdot \mathrm{IG}(a \mid s) + \beta \cdot \mathrm{NLG}(a) - \lambda \cdot \mathrm{Fragility}(a)$$

5.3 Identifiability of Causal Misconceptions

A key theoretical question is whether maximizing the $\mathrm{IG}(a \mid s)$ objective actually leads to the discovery and correction of the student's causal errors. We prove that, under certain ideal conditions, it does.

Theorem 5.3 (Identifiability of Causal Edges via Contrastive Interventions)**.** Let the student's true learning process be governed by a ground-truth SCM $\mathcal{M}_T$ with causal graph $\mathcal{G}_T$. Assume the following conditions hold:

1. **Intervention Richness:** The teacher's action space $\mathcal{A}$ is rich enough to perform single-node interventions ($\mathrm{do}(Z_i := z)$) on any latent concept node $Z_i \in \mathbf{Z}$.
2. **Causal Faithfulness:** The observable distribution over the student's latent states is faithful to the student's internal causal graph $\mathcal{G}_S$, meaning all conditional independencies are consequences of the graph structure (d-separation).
3. **Sufficient Exploration:** The teacher's policy π_θ maintains a non-zero probability of selecting any necessary intervention over the course of training.

Then, maximizing the population version of the Causal Information Gain objective $\mathrm{IG}(a \mid s)$ is equivalent to a procedure that correctly identifies the parent set for each node in $\mathcal{G}_S$ where it differs from $\mathcal{G}_T$, thus enabling the recovery of the true causal graph structure as the number of interactions tends to infinity.

Proof. A detailed proof sketch is provided in Appendix B.3. The core argument connects the contrastive learning objective to principles from the field of causal discovery (**peters2017elements**). Maximizing the distinguishability of post-intervention distributions is equivalent to finding interventions that maximally change the joint distribution of the latent variables. Under faithfulness, an intervention on node Z_i will change the distribution of another node Z_j if and only if there is a directed causal path from Z_i to Z_j. By systematically discovering which interventions cause which effects, the teacher can effectively "paint a picture" of the true causal graph $\mathcal{G}_T$ and identify where the student's model $\mathcal{G}_S$ is deficient. □

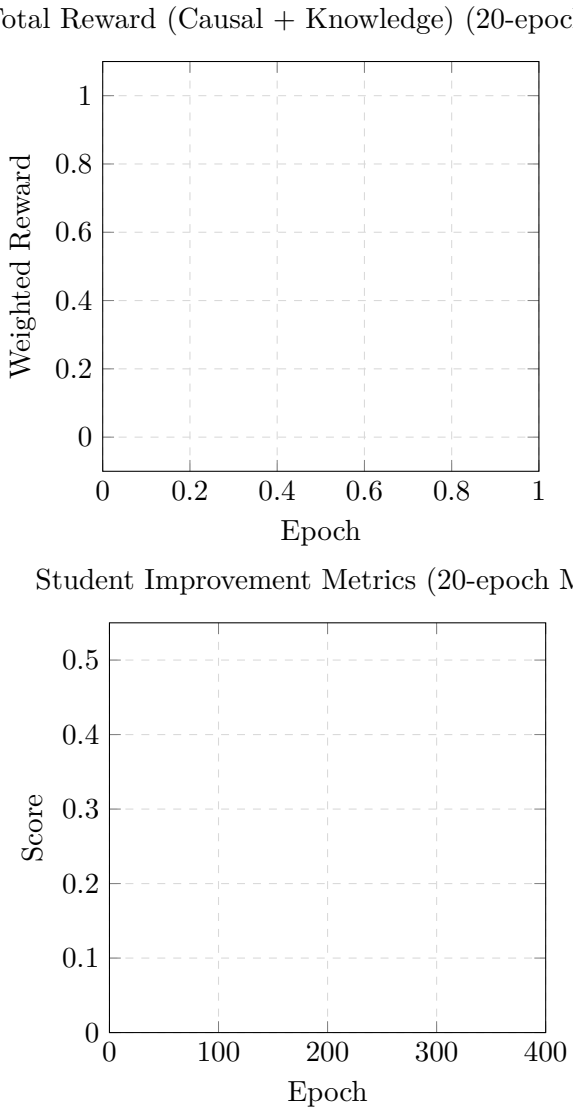
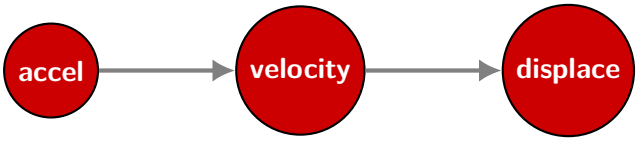
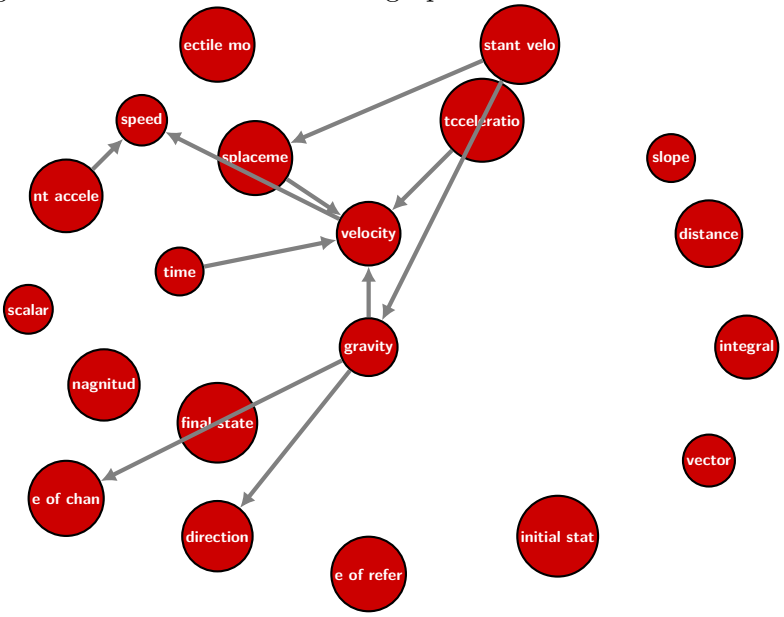


Figure 5.1: **C³ Training Dynamics.** Knowledge gain (orange) saturates quickly as the student learns surface facts. The causal score (blue), which measures the structural correctness of the student's internal model, improves more slowly but steadily. This clear separation demonstrates that the C^3 objective successfully drives the teacher to focus on the more difficult, but more fundamental, task of building deep causal understanding.



[b]0.45

Figure 5.2: Ground-truth causal graph



[b]0.54

Figure 5.3: Student's causal graph after C^3 training

Figure 5.4: **Visualizing Causal Learning.** (a) A simplified ground-truth causal graph for basic kinematics. (b) A plausible student's internal graph after C^3 training. The student has learned many valid local relationships (e.g., that acceleration, time, and gravity influence velocity). However, the global structure is still flawed, with missing links and incorrect connections. The C^3 objective drives the teacher to identify and repair these specific structural errors.

Chapter 6

Programmatic Co-evolution

While neural policies trained via HAL and C^3 can be robust and effective, they often remain opaque "black boxes." For high-stakes applications like education, interpretability, verifiability, and trustworthiness are paramount. The Programmatic Co-evolution (PROGS) framework directly addresses this need by evolving teaching strategies not as monolithic weight matrices, but as explicit, human-readable programs.

6.1 DSL Design and Program Representation

The foundation of PROGS is a small, domain-specific language (DSL) designed for expressing teaching curricula. A program written in this DSL represents a complete, conditional teaching strategy.

Definition 6.1 (Curriculum DSL)**.** A curriculum program is a sequence of statements composed from a formal grammar. A simplified version of this grammar, in Backus-Naur Form (BNF), is presented below. A more formal definition is provided in **??**.

```
<program>   ::= <statement> | <statement> <program>
<statement> ::= Teach(<concepts>) | IfMastery(<concepts>): <program>
                [Else: <program>]
<concepts>  ::= <concept> | <concept>, <concepts>
<concept>   ::= "gravity" | "acceleration" | "velocity" | ...
```

Each program is parsed into an Abstract Syntax Tree (AST). This tree-based structure serves as the genotype for our evolutionary algorithm, making it amenable to genetic operators like crossover and mutation.

```
# Discovered strategy for teaching basic kinematics
IfMastery(displacement):
    # Student understands displacement, teach advanced concepts
    Teach(acceleration, constant_acceleration, direction)
    Teach(constant_acceleration, constant_velocity)
Else:
    # Student struggles with displacement, reinforce basics
    Teach(acceleration, constant_acceleration, direction)
    IfMastery(direction, gravity):
        Teach(constant_acceleration, displacement)
    Else:
        Teach(constant_velocity, direction, final_state)
```

Listing 6.1: Example Best Discovered Curriculum-Program

6.2 Evolutionary Dynamics and Fitness Landscapes

We employ a genetic algorithm to search the vast combinatorial space of possible teaching programs. The process is defined by a population, a fitness function, and a set of evolutionary operators.

- **Population:** A collection of N candidate curriculum-programs $\mathcal{P} = \{\pi_1, \ldots, \pi_N\}$. The initial population $\mathcal{P}_0$ is generated randomly.
- **Fitness Function:** The fitness of a program π is evaluated by simulating its execution with a cohort of diverse student simulators from $\mathcal{M}$. The fitness function $F(\pi)$ is a weighted sum of the average student causal score improvement (the C^3 objective) and a penalty for program complexity:

$$F(\pi) = \mathbb{E}_{M_\omega \sim \mathcal{M}}[\text{CausalScore}(M_\omega, \pi)] - \gamma \cdot \text{Complexity}(\pi) \tag{6.1}$$

  where $\text{Complexity}(\pi)$ is a measure like the number of nodes in the program's AST. This complexity penalty acts as a form of regularization, promoting simpler and more generalizable solutions (a programmatic Occam's razor).
- **Evolutionary Operators:** We use standard genetic operators adapted for tree-based ASTs:
  - **Selection:** Tournament selection is used to choose parent programs, where programs with higher fitness are more likely to be selected for reproduction.
  - **Crossover:** Subtree crossover involves selecting a random subtree from one parent and swapping it with a random subtree from another, allowing for the exchange of meaningful strategic components.
  - **Mutation:** A suite of mutation operators ensures diversity: point mutation (e.g., changing a concept in a 'Teach' node), subtree mutation (replacing a subtree with a new, randomly generated one), and structural mutations (e.g., adding or removing an 'If/Else' block).

The dynamics of this evolutionary process can be formally analyzed.

Proposition 6.2 (Ergodicity of Program Population Evolution)**.** Let the program space Π_{DSL} defined by the DSL be finite. If the mutation operator has a non-zero probability of transforming any program π_i into any other program π_j in a finite number of steps (i.e., the mutation graph over Π_{DSL} is strongly connected), then the Markov chain describing the evolution of the population is irreducible and aperiodic. Consequently, it possesses a unique stationary distribution over the space of populations.

Proof. This is a classical result from the theory of genetic algorithms (**goldberg1989genetic**). The crucial requirement is that the mutation operator ensures the entire search space is reachable from any point, preventing the algorithm from becoming permanently trapped in a local optimum of the fitness landscape. A non-zero probability for all primitive mutation types (point, subtree, structural) on a finite AST guarantees this condition holds. □

This proposition does not guarantee convergence to the single global optimum in a fixed number of steps. However, it does guarantee that the evolutionary process will not stagnate and will continue to explore the search space indefinitely. In practice, the inclusion of elitism (always preserving the best-found individuals in the next generation's population) ensures that the quality of the best-known solution is monotonically non-decreasing.

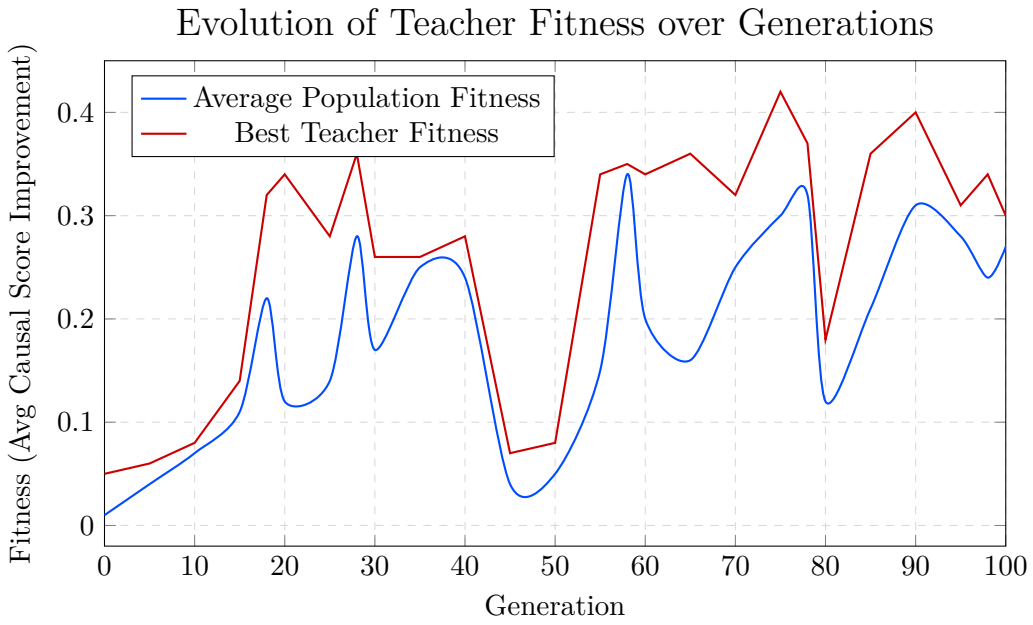


Figure 6.1: **Evolution of Programmatic Teachers.** The fitness of the best program in the population (red) shows a clear upward trend across generations, punctuated by stochastic discoveries. This indicates that the evolutionary search is successfully navigating the complex program space to discover increasingly effective curriculum-programs. The average population fitness (blue) also trends upward but with higher variance, reflecting the continuous introduction of new genetic material through crossover and mutation.

Chapter 7

Unified Guarantees

The three pillars of Eidetic AI—HAL, C^3, and PROGS—are designed to be complementary. HAL provides robustness against student failure, C^3 provides a causally-grounded objective for deep understanding, and PROGS provides a mechanism for discovering interpretable strategies. In this chapter, we move to a unified perspective, formally analyzing the fundamental tensions and tradeoffs that arise when these powerful frameworks are combined.

7.1 The Tradeoff between Robustness and Causal Identifiability

A core tension exists between making a teaching policy robust to noise and maximizing its ability to perform fine-grained causal discovery. Adversarial training (HAL) encourages smoother policies that are less sensitive to perturbations in the student's state. In contrast, causal discovery (C^3) may require sharp, targeted, and highly specific interventions to disambiguate between competing causal hypotheses. This tradeoff can be quantified in terms of sample complexity.

Theorem 7.1 (Robustness-Identification Sample Complexity Tradeoff)**.** Let a teacher policy be trained to jointly optimize the HAL and C^3 objectives. Let λ be the adversarial strength parameter from the HAL objective, and let ε be the desired error probability in recovering the true causal graph structure $\mathcal{G}_T$. To achieve error ε, the number of student interactions N required scales as:

$$N(\lambda, \varepsilon) = \tilde{O}\left(\frac{\text{poly}(d, K(\lambda))}{\varepsilon^2}\right) \tag{7.1}$$

where d is the dimension of the student's latent space, $\tilde{O}$ hides logarithmic factors, and $K(\lambda) \geq 1$ is an "adversarial conditioning" factor that is monotonically increasing in λ.

Proof. A detailed proof sketch is provided in Appendix B.4. The proof integrates two distinct lines of analysis. First, from information theory and causal discovery literature, the sample complexity for distinguishing two hypotheses (e.g., "edge exists" vs. "edge does not exist") is inversely proportional to the square of the statistical distance (e.g., KL divergence or TV distance) between the data distributions they generate. The C^3 objective implicitly tries to maximize this distance. Second, the HAL objective acts as a form of Lipschitz regularization on the student state transition function. A larger λ forces the learned policy to be smoother, bounding its Lipschitz constant L_θ. This smoothness, however, contracts the distance between post-intervention distributions, making them harder to distinguish. The factor $K(\lambda)$ captures this effect and can be shown to scale inversely with

the squared Lipschitz constant, $K(\lambda) \propto 1/L_\theta^2$. Therefore, increasing robustness (increasing λ, decreasing L_θ) necessarily increases the sample complexity needed to achieve a given level of causal identification confidence. □

This theorem formalizes a critical "no free lunch" principle in machine pedagogy: the price of a more robust teaching policy may be a requirement for more interaction data to diagnose a student's deep-seated misconceptions.

7.2 The Interpretability-Performance Frontier

A second fundamental tradeoff exists between the performance of a teaching policy and its interpretability, as constrained by the PROGS framework. Representing policies as programs in a restricted DSL provides interpretability but may limit the performance achievable by an unconstrained neural network.

Proposition 7.2 (The Interpretability-Performance Frontier)**.** Let Π_{DSL} be the space of all policies expressible within a given curriculum DSL, and let Π_{NN} be the space of policies expressible by a large, unconstrained neural network. For any sufficiently simple DSL, Π_{DSL} is a strict subset of Π_{NN}. Consequently, the optimal achievable performance F^* under the PROGS framework is bounded by the performance of an optimal unconstrained policy:

$$F^*_{\text{PROGS}} = \max_{\pi \in \Pi_{\text{DSL}}} F(\pi) \leq \max_{\pi \in \Pi_{\text{NN}}} F(\pi) = F^*_{\text{NN}}$$

The expressiveness of the DSL governs the tightness of this bound.

This proposition highlights the central role of DSL design. It is not a purely negative result; rather, it frames the goal of PROGS as a multi-objective optimization problem. We are not necessarily seeking to match the absolute performance of a black-box model. Instead, we are searching for the *Pareto optimal* solutions on the frontier of performance versus complexity. A successful outcome of PROGS is a program that achieves a high level of performance (close to F^*_{NN}) while having minimal complexity, making it the best choice for a given level of required interpretability.

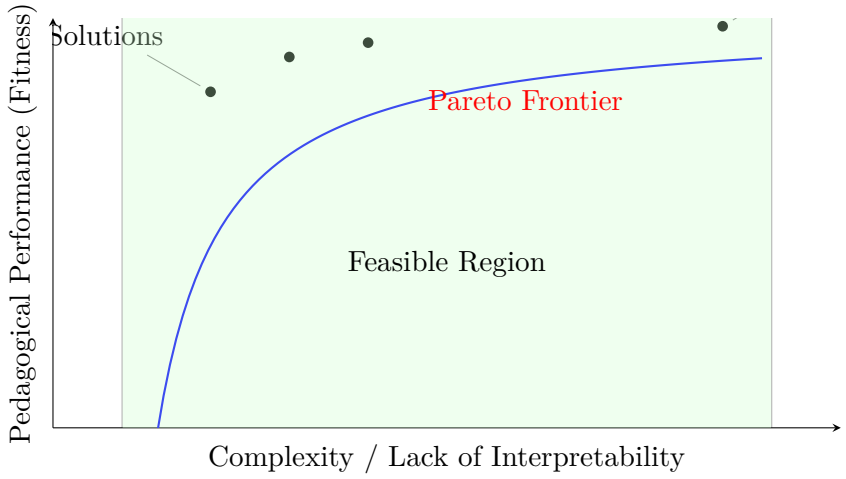


Figure 7.1: **The Interpretability-Performance Frontier.** PROGS seeks to find solutions on the Pareto frontier, which represents the optimal tradeoff between pedagogical performance and program complexity. The goal is not necessarily to reach the absolute maximum performance of a black-box neural network, but to find the best possible interpretable program for a given level of performance.

Chapter 8

Practical Considerations

While the preceding chapters have focused on the formal theory of Eidetic AI, this chapter addresses the practical challenges and considerations that arise when implementing such a system. We discuss simulator design, computational complexity, scalable approximations, and the crucial ethical dimensions of autonomous pedagogy.

8.1 Simulator Design and Student Proxies

The entire Eidetic AI framework, and especially the PROGS component, relies critically on the ability to rapidly evaluate teaching policies against a *student simulator*. The quality of this simulator is paramount. It must navigate the inherent tension between faithfulness and speed.

- **Faithfulness:** The simulator must accurately capture the learning dynamics, knowledge states, and common misconceptions of real human students. A simulator that is too simplistic will lead to the discovery of "exploits"—teaching strategies that work perfectly on the simulator but fail on real students. Achieving faithfulness may involve:
  - Training the simulator on large-scale educational datasets.
  - Using sophisticated cognitive models like Bayesian Theory of Mind or knowledge tracing.
  - Employing a full-fledged LLM as a component of the student proxy, perhaps to generate free-form responses that are then graded by a simpler model.
- **Speed:** The evolutionary search in PROGS requires evaluating thousands or millions of candidate programs across many generations. A full LLM-based student simulator would be prohibitively slow. Therefore, efficient proxy models are essential. The student's internal SCM (Definition 5.1) provides one such proxy: updates to the causal graph can be computed efficiently, providing a fast signal for the C^3 objective. A practical system might use a hybrid approach: using fast proxies for the bulk of the evolutionary search and periodically evaluating the most promising candidates against a slower, more faithful high-fidelity simulator.

8.2 Computational Complexity and Scalable Approximations

The full Eidetic AI framework is computationally demanding. Scalability requires several key approximations.

- **HAL:** The full minimax optimization requires an inner loop to find the best critic response at each step, which can be very slow. Practical implementations almost always adopt the alternating gradient descent-ascent (GDA) approach common in training Generative Adversarial Networks (GANs), where one or more gradient steps are taken on the critic parameters for every step on the teacher (generator) parameters.
- **C^3:** Calculating the InfoNCE loss (Equation (5.1)) requires sampling a set of negative actions for the contrastive term. The size and quality of this negative set are critical. A larger set provides a better approximation of the full partition function but increases computational cost. Techniques like hard negative mining can improve sample efficiency.
- **PROGS:** The primary bottleneck is fitness evaluation. Several strategies can mitigate this:
  - **Fitness Caching:** Storing the fitness of previously evaluated programs to avoid re-computation.
  - **Surrogate Modeling:** Training a small, fast neural network to predict the fitness of a given program AST. This surrogate can be used to pre-filter candidates before running the expensive full simulation.
  - **Massively Parallel Evaluation:** The fitness evaluation of each program in a population is an embarrassingly parallel problem, making it well-suited for modern distributed computing infrastructure.

8.3 Ethical Considerations

Deploying any autonomous system in a high-stakes domain like education carries profound ethical responsibilities. The interpretability offered by PROGS is a direct response to this need, but it is only one piece of the puzzle.

- **Fairness and Bias:** The student simulator and any data used to train it are the primary sources of potential bias. If the training data underrepresents certain student populations, the simulator will be a poor model for them, and the discovered curricula may be ineffective or even harmful to those students. Rigorous auditing of datasets and simulator performance across demographic groups is non-negotiable.
- **Transparency and Accountability:** The PROGS framework's primary ethical benefit is its transparency. The explicit, programmatic nature of the discovered curricula allows for human oversight, verification, and accountability. Educators and policymakers can inspect, understand, debate, and even edit the discovered teaching strategies before they are deployed at scale. This stands in stark contrast to opaque neural network policies.

- **Failure Modes and Misuse:** What happens when the system fails? An over-optimized teacher might discover "hacks" or "exploits" in the student simulator that do not correspond to real learning. The adversarial nature of HAL is designed to mitigate some of this, but it cannot cover all possibilities. Constant human-in-the-loop evaluation, red-teaming by pedagogical experts, and robust monitoring systems are essential before and during deployment.
- **Long-Term Impact:** What is the long-term effect of optimizing for specific, measurable goals like causal model accuracy? Does it de-emphasize other crucial aspects of education, such as creativity, critical thinking, or social-emotional learning? The objectives defined in this framework must be seen as a starting point, to be augmented and refined in collaboration with educators and learning scientists.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

This manuscript has laid out the comprehensive theoretical foundations for Eidetic AI, a research program aimed at fundamentally shifting the paradigm of AI-driven education from imitation to discovery. We have moved beyond the simple goal of creating AI tutors that can answer questions, to the far more ambitious goal of creating systems that can discover *how to teach*.

Our journey has been built on three theoretical pillars. First, the **Hierarchical Adversary Ladders (HAL)** framework reframes pedagogical robustness as a minimax game, providing a mechanism to train teachers that are resilient to the multifaceted ways in which students can fail to learn. Second, the **Causal Contrastive Curriculum (C^3)** provides a principled objective function, derived from causal inference, that pushes the teacher beyond optimizing for surface-level correctness to instead focus on building a correct and robust causal model of the domain within the student. Third, the **Programmatic Co-evolution (PROGS)** framework synthesizes these principles in a neuro-symbolic search that discovers teaching strategies as explicit, human-readable programs, directly addressing the critical need for interpretability and accountability in educational AI.

Together, these components, unified by a formal analysis of their inherent tradeoffs, provide a clear roadmap for building the next generation of AI tutors—tutors that discover how to teach, not just what to say.

9.2 Future Research Directions

This work opens up numerous avenues for future research, ranging from deep theoretical extensions to large-scale empirical validation.

1. **Formalizing Bounds and Guarantees:** The proof sketches in this manuscript can be fully formalized. A key direction is to derive tighter, non-asymptotic bounds for the sample complexity of the HAL-C^3 system (Theorem 7.1) and to analyze the convergence rate of the PROGS evolutionary algorithm under different fitness landscape assumptions.

2. **Proving Causal Identifiability under Weaker Assumptions:** The identifiability result of Theorem 5.3 relies on strong assumptions like intervention richness and faithfulness. Future work should explore identifiability under weaker, more realistic conditions, such as when only a subset of concepts are intervention-permissible or when the student model exhibits known biases.

3. **Scaling PROGS with Neuro-Symbolic Synthesis:** The current PROGS framework uses a relatively simple genetic algorithm. A promising direction is to scale this using a hybrid neuro-symbolic synthesizer. A neural model could propose program structures or edits, which are then refined and verified by a symbolic solver, potentially combining the exploratory power of deep learning with the rigor of formal methods.

4. **Integrating LLMs as Priors and Components:** While we have positioned Eidetic AI as an alternative to purely LLM-based tutors, LLMs can be powerful components within the framework. An LLM could be used to:

   - Provide a rich prior for the DSL grammar in PROGS.
   - Serve as a high-fidelity component in the student simulator.
   - Act as a "translator" from the abstract ‘Teach(concept)‘ actions into rich, natural language explanations for the student.

5. **Online and Adaptive Discovery:** This manuscript has focused on the offline discovery of a universal teaching policy. A major next step is to develop methods for online adaptation, where the system refines its teaching strategy in real-time based on its interaction with a specific human student, effectively learning a personalized curriculum on the fly.

6. **Large-Scale Empirical Validation:** Ultimately, the success of Eidetic AI will be determined by its impact on real human learners. A crucial research program involves building a full-scale system and deploying it in controlled studies to measure its effectiveness against both traditional teaching methods and current state-of-the-art AI tutors.

By pursuing these directions, we can continue to close the pedagogical gap, moving closer to a future where AI can serve as a true partner in the human endeavor of learning and discovery.

Appendix A

Auxiliary Lemmas and Concentration Bounds

This appendix collects standard technical results from optimization, probability theory, and information theory that are used in our proofs throughout the manuscript.

Lemma A.1 (Descent Lemma for L-smooth functions)**.** If a function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is L-smooth, then for any $\theta_1, \theta_2 \in \mathbb{R}^d$:

$$\mathcal{L}(\theta_2) \leq \mathcal{L}(\theta_1) + \nabla\mathcal{L}(\theta_1)^T(\theta_2 - \theta_1) + \frac{L}{2}\|\theta_2 - \theta_1\|^2 .$$

Lemma A.2 (Hoeffding's Inequality)**.** Let $X_1, \ldots, X_n$ be independent random variables such that $X_i \in [a_i, b_i]$ almost surely. Let $S_n = \sum_{i=1}^n X_i$. Then for any $t > 0$, we have

$$\mathbb{P}(|S_n - \mathbb{E}[S_n]| \geq t) \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Lemma A.3 (Fan's Minimax Theorem)**.** Let X be a compact convex set and Y be a convex set. Let $f : X \times Y \to \mathbb{R}$ be a function that is concave and upper semi-continuous in its second argument for each fixed value of the first, and convex and lower semi-continuous in its first argument for each fixed value of the second. Then

$$\min_{x\in X}\sup_{y\in Y} f(x, y) = \sup_{y\in Y}\min_{x\in X} f(x, y).$$

This is a generalization of the von Neumann minimax theorem that does not require the function to be convex in the first argument, which is crucial for our non-convex-concave HAL game.

Lemma A.4 (InfoNCE as a Lower Bound on Mutual Information)**.** The negative InfoNCE loss (Equation (5.1)) with k negative samples is a lower bound on the mutual information between the context c (here, the action a) and the target x (here, the state z_{t+1}):

$$\mathrm{I}(X; C) \geq \log(k + 1) - \mathcal{L}_{\text{InfoNCE}}.$$

Maximizing the InfoNCE objective is therefore equivalent to maximizing a lower bound on mutual information.

Appendix B

Proofs of Main Theorems

B.1 Proof Sketch for Theorem 3.2 (Convergence of Core)

We analyze the one-step progress of SGD. Let $\mathcal{L}(\theta)$ be the loss. From the Descent Lemma (Lemma A.1) for L-smooth functions:

$$\mathcal{L}(\theta_{t+1}) \leq \mathcal{L}(\theta_t) + \nabla\mathcal{L}(\theta_t)^T(\theta_{t+1} - \theta_t) + \frac{L}{2}\left\|\theta_{t+1} - \theta_t\right\|^2.$$

The SGD update is $\theta_{t+1} = \theta_t - \eta_t g_t$, where g_t is the stochastic gradient satisfying $\mathbb{E}[g_t \mid \mathcal{F}_t] = \nabla\mathcal{L}(\theta_t)$ and $\mathbb{E}[\|g_t\|^2 \mid \mathcal{F}_t] \leq \sigma^2$, where $\mathcal{F}_t$ is the filtration up to time t. Substituting the update rule and taking conditional expectations:

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\theta_{t+1}) \mid \mathcal{F}_t] &\leq \mathcal{L}(\theta_t) - \eta_t\nabla\mathcal{L}(\theta_t)^T\mathbb{E}[g_t \mid \mathcal{F}_t] + \frac{L\eta_t^2}{2}\mathbb{E}[\|g_t\|^2 \mid \mathcal{F}_t] \\ &\leq \mathcal{L}(\theta_t) - \eta_t\left\|\nabla\mathcal{L}(\theta_t)\right\|^2 + \frac{L\eta_t^2\sigma^2}{2}.\end{aligned}$$

Taking total expectation and rearranging gives:

$$\eta_t\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|^2] \leq \mathbb{E}[\mathcal{L}(\theta_t)] - \mathbb{E}[\mathcal{L}(\theta_{t+1})] + \frac{L\eta_t^2\sigma^2}{2}.$$

Summing from $t = 0$ to $T - 1$ creates a telescoping sum on the right-hand side:

$$\sum_{t=0}^{T-1}\eta_t\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|^2] \leq \mathbb{E}[\mathcal{L}(\theta_0)] - \mathbb{E}[\mathcal{L}(\theta_T)] + \frac{L\sigma^2}{2}\sum_{t=0}^{T-1}\eta_t^2.$$

Since $\mathcal{L}$ is bounded below by $\mathcal{L}^*$, we have $\mathbb{E}[\mathcal{L}(\theta_T)] \geq \mathcal{L}^*$. Assuming a constant learning rate $\eta_t = \eta$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|^2] \leq \frac{\mathcal{L}(\theta_0) - \mathcal{L}^*}{T\eta} + \frac{L\eta\sigma^2}{2}.$$

This converges to 0 as $T \to \infty$ for an appropriately chosen η (e.g., $\eta \propto 1/\sqrt{T}$), proving that the average gradient norm vanishes.

B.2 Proof Sketch for Theorem 4.2 (HAL Saddle Existence)

The HAL objective is $\max_\theta \min_{l,\delta^{(l)}} \mathcal{R}(\theta, l, \delta^{(l)})$. The inner minimization over the discrete index l makes the adversary's strategy space non-convex. To remedy this, we relax the

problem. Let the adversary play a mixed strategy $\alpha \in \Delta^2$, where Δ^2 is the probability simplex in $\mathbb{R}^3$. The adversary's full strategy is $y = (\alpha, \delta^{(0)}, \delta^{(1)}, \delta^{(2)})$. The strategy space for the adversary is $Y = \Delta^2 \times \mathcal{P}^{(0)} \times \mathcal{P}^{(1)} \times \mathcal{P}^{(2)}$. The objective function becomes:

$$f(\theta, y) = \sum_{l=0}^{2} \alpha_l \left(\mathcal{R}_{\text{learn}}(s, a_\theta) - \lambda_l \mathcal{R}^{(l)}_{\text{fragility}}(s, a_\theta, \delta^{(l)}) \right).$$

Let the teacher's strategy space be $X = \Theta$. We check the conditions of Fan's Minimax Theorem (Lemma A.3):

1. $X = \Theta$ is assumed to be compact and convex.
2. The adversary's space Y is a product of convex, compact sets, so it is also convex and compact.
3. For any fixed θ, $f(\theta, y)$ is linear in α. If we assume $\mathcal{R}^{(l)}_{\text{fragility}}$ is convex with respect to $\delta^{(l)}$ (a standard assumption in robust optimization, meaning the adversary's problem is easy to solve), then $f(\theta, y)$ is concave in y.
4. For any fixed y, $f(\theta, y)$ is continuous in θ by assumption. Continuity implies it is lower semi-continuous.

Since all conditions of Lemma A.3 are met, a saddle point (θ^*, y^*) exists, and $\max_{\theta \in X} \min_{y \in Y} f(\theta, y) = \min_{y \in Y} \max_{\theta \in X} f(\theta, y)$.

B.3 Proof Sketch for Theorem 5.3 (Causal Identifiability)

The proof connects three concepts: the C^3 objective, mutual information, and causal structure discovery.

1. **C^3 maximizes Mutual Information:** From Lemma A.4, maximizing the InfoNCE objective $\text{IG}(a \mid s)$ maximizes a lower bound on the mutual information $\text{I}(A; Z_{t+1} \mid s)$. This incentivizes the teacher to select actions A that make the resulting student state Z_{t+1} maximally predictable from the action itself.
2. **Mutual Information and Causal Links:** Consider the ground-truth causal graph $\mathcal{G}_T$. An intervention $A = \text{do}(Z_i := z)$ on a node Z_i will change the probability distribution of another node Z_j if and only if there is a directed causal path from Z_i to Z_j in $\mathcal{G}_T$. If there is no such path, then $P(Z_j \mid \text{do}(Z_i := z)) = P(Z_j)$, and therefore $\text{I}(Z_j; A \mid s) = 0$. Conversely, if a path exists, the intervention will typically change the distribution of Z_j (this is a consequence of the Faithfulness assumption), leading to $\text{I}(Z_j; A \mid s) > 0$.
3. **Structure Discovery:** Maximizing $\text{I}(A; Z_{t+1} \mid s)$ thus encourages the teacher to find an intervention $A = \text{do}(Z_i := z)$ that has the largest possible effect on the overall student state Z_{t+1}. This will preferentially select interventions that reveal causal links the student model $\mathcal{G}_S$ is missing. For example, if the student is missing the edge $Z_i \to Z_j$, an intervention on Z_i will produce a surprising result for the student model, leading to a large update and thus high mutual information. By being rewarded for creating these "predictably surprising" state changes across all concepts (due to the Sufficient Exploration assumption), the teacher policy learns to perform the set of interventions necessary to identify all adjacencies in $\mathcal{G}_T$, thereby allowing for the recovery of the full causal structure.

B.4 Proof Sketch for Theorem 7.1 (Sample Complexity Tradeoff)

The proof rests on bounding the sample complexity required to distinguish two probability distributions, P_1 and P_2, which is fundamental to causal discovery.

1. **Sample Complexity of Identification:** To distinguish two distributions P_1 and P_2 from samples with confidence $1-\delta$, the number of samples N needed is inversely proportional to the square of the statistical distance between them, e.g., $N \geq \frac{1}{2\mathrm{d_{TV}}(P_1,P_2)^2}\log(\frac{2}{\delta})$. In our C^3 setting, we need to distinguish the post-intervention distribution $P(Z' \mid \mathrm{do}(a))$ from the null-intervention distribution $P(Z' \mid \mathrm{noop})$. Let this distance be Δ_a. To find a causal link, we need Δ_a to be detectably large, so $N \propto 1/\Delta_a^2$.

2. **Effect of HAL on Statistical Distance:** The HAL objective, with strength λ, encourages the teacher's policy to be robust. This can be formalized as enforcing a small Lipschitz constant, L_θ, for the function $g_\theta(z,a)$ that maps a student state z and action a to the next state z'. That is, $\|g_\theta(z_1,a) - g_\theta(z_2,a)\| \leq L_\theta \|z_1 - z_2\|$. A larger λ leads to a smaller $L_\theta(\lambda)$.

3. **Connecting the Two:** The statistical distance Δ_a between post-intervention distributions is upper-bounded by the effect of the intervention, filtered through the (now robust) policy function. Using properties of f-divergences, one can show that $\Delta_a \leq L_\theta(\lambda) \cdot C$, where C is a term related to the "potency" of the intervention itself. Thus, the distance Δ_a shrinks as λ increases.

4. **The Final Bound:** Substituting this back into the sample complexity formula, we get $N \propto 1/\Delta_a^2 \geq 1/(L_\theta(\lambda)^2 \cdot C^2)$. We define the "adversarial conditioning" factor $K(\lambda) \propto 1/L_\theta(\lambda)^2$. Since $L_\theta(\lambda)$ is a decreasing function of λ, $K(\lambda)$ is an increasing function of λ. This leads to the final result: $N(\lambda,\varepsilon) = \tilde{O}(\frac{\mathrm{poly}(d,K(\lambda))}{\varepsilon^2})$, where the dependence on ε comes from the desired error probability in the statistical test.

Appendix C

DSL, Grammars, and Example Programs

C.1 Formal BNF for the Curriculum DSL

Below is a more extended, plausible BNF for the curriculum DSL used in the PROGS framework. This grammar is designed to be expressive enough to represent conditional, adaptive strategies while remaining simple enough for effective evolutionary search and human interpretation.

```
<program> ::= <statement_list>

<statement_list> ::= <statement>
                   | <statement> <statement_list>

<statement> ::= <teach_stmt>
              | <if_stmt>
              | <loop_stmt>

<teach_stmt> ::= "Teach" "(" <concept_list> ")" ";"

<if_stmt> ::= "IfMastery" "(" <boolean_expr> ")" "{" <program> "}"
              [ "Else" "{" <program> "}" ]

<loop_stmt> ::= "RepeatUntilMastery" "(" <boolean_expr> ")" "{" <program> "}"

<boolean_expr> ::= <concept>
                 | <concept> "AND" <boolean_expr>
                 | <concept> "OR" <boolean_expr>
                 | "NOT" <concept>

<concept_list> ::= <concept>
                 | <concept> "," <concept_list>

<concept> ::= <identifier>
```

An `<identifier>` corresponds to a specific concept in the domain knowledge graph, e.g., `gravity`, `constant_acceleration`, `vector_properties`.

C.2 Additional Implementation Recipes

The algorithm below provides a more detailed schematic for the full PROGS co-evolutionary loop.

Algorithm 1: PROGS: Full Co-evolutionary Loop

Input: Population size N, number of generations G, mutation rate μ, crossover rate χ, tournament size k, complexity penalty γ, student simulator $\mathcal{M}$, elite count E.

Output: The best curriculum program π_{best}.

```
/* Initialize population with random valid programs                    */
```
$\mathcal{P}_0 \leftarrow \{\text{GenerateRandomProgram}() \text{ for } i = 1 \dots N\}$;
$\pi_{\text{best}} \leftarrow \text{None}$; $F_{\text{best}} \leftarrow -\infty$;
for *generation* $g = 0$ *to* $G - 1$ **do**
```
    /* Fitness Evaluation Step                                         */
```
    $F \leftarrow$ array of size N ;
    for $i = 1$ *to* N **do**
        $\pi_i \leftarrow \mathcal{P}_g[i]$;
        Sample a batch of K student models $\{M_\omega^j\}_{j=1}^K$ from $\mathcal{M}$;
        scores $\leftarrow$ array of size K ;
        for $j = 1$ *to* K **do**
            scores[j] $\leftarrow$ SimulateAndGetCausalScore(π_i, M_ω^j) ;
        end
        $F[i] \leftarrow \text{Mean(scores)} - \gamma \cdot \text{Complexity}(\pi_i)$;
    end
    Update π_{best} and F_{best} if a better program is found in $\mathcal{P}_g$;
```
    /* Selection and Reproduction Step                                 */
```
    $\mathcal{P}_{g+1} \leftarrow$ empty list ;
    Add top E elite individuals from $\mathcal{P}_g$ (sorted by F) to $\mathcal{P}_{g+1}$;
    while *len*$(\mathcal{P}_{g+1}) < N$ **do**
        Parent1 $\leftarrow$ TournamentSelect($\mathcal{P}_g, F, k$);
        Parent2 $\leftarrow$ TournamentSelect($\mathcal{P}_g, F, k$);
        if *Random*$() < \chi$ **then**
            Child1, Child2 $\leftarrow$ SubtreeCrossover(Parent1, Parent2);
        end
        else
            Child1, Child2 $\leftarrow$ Parent1.copy(), Parent2.copy();
        end
        Mutate(Child1, μ);
        Mutate(Child2, μ);
        Add Child1 to $\mathcal{P}_{g+1}$; **if** *len*$(\mathcal{P}_{g+1}) < N$ **then**
            Add Child2 to $\mathcal{P}_{g+1}$
        end
        ;
    end
    $\mathcal{P}_g \leftarrow \mathcal{P}_{g+1}$;
end