Self-Verifying Synthetic Data Generation for AGI Development

A Revolutionary Approach to Artificial General Intelligence Through Automated Ground Truth Accumulation

Version: 3.0

Date: November 29, 2025

Author: Rujing Tang

Contact: RJ@qqqtech.com

Status: Public Release

Executive Summary

We present a pathway to Artificial General Intelligence that breaks free from the fundamental limitations constraining current AI development. The bottleneck is not computational power or model architecture—it is data quality. Large language models today train on billions of unverified internet documents, learning patterns of both truth and falsehood with equal confidence. This creates sophisticated prediction engines that cannot reliably distinguish fact from fiction.

Our approach transforms this paradigm through a simple insight: AGI emerges when comprehensive verified ground truth combines with effective reasoning mechanisms. We express this as AGI = Ground_Truth_Dataset + Reasoning_Logic. The challenge lies in accumulating verified ground truth at the scale needed for AGI—billions of examples across all domains of human knowledge—without the human labeling bottleneck that makes traditional supervised learning impractical.

A critical insight often overlooked is that asking the right question with the right prompt determines what knowledge can be acquired. Human users struggle with question formulation—they ask "how do I make my code faster?" without specifying what code, what bottleneck, or what constraints apply. This limitation multiplies when AI generates questions for its own training. Poor questions yield vague, unverifiable answers even with perfect answer generation. Our approach explicitly trains question quality through analyzing millions of human-AI interactions, identifying patterns that distinguish high-value questions from low-value ones. The AI learns not just to answer but to ask, targeting knowledge gaps, including necessary context, probing boundary conditions, and enabling verification. Question quality compounds across generations: poor questions yield thirty percent useful training data, medium questions yield sixty percent, high-quality questions yield ninety percent. Over three generations, this 3x multiplier in useful data per question dramatically accelerates discovery capability.

The solution is self-verifying synthetic data generation. Rather than waiting for humans to ask questions and label answers, we generate millions of synthetic questions systematically across parameter spaces, have AI answer them, verify the answers through multiple automated methods, and train exclusively on verified correct answers. This creates a virtuous cycle where each model generation improves both its answer quality and its verification capabilities, leading to compounding improvement over time.

The Core Innovation: Human-Guided Discovery Through Usage Patterns

The self-discovery process is not random exploration. Instead, it follows the actual trajectory of human AI usage. When millions of users interact with AI systems, their questions reveal what knowledge humans actually seek, what domains matter most, and what edge cases arise in practice. This usage data becomes a natural curriculum that guides synthetic question generation.

Real-world usage patterns tell us where to focus verification efforts. If users frequently ask about medical diagnoses, we generate comprehensive synthetic questions in medicine. If users struggle with coding problems, we expand our code generation and verification. If users hit edge cases in engineering calculations, we systematically explore those boundary conditions. The AI thus discovers along the path of human need rather than wandering randomly through possibility space.

This human-guided discovery solves a critical problem: not all knowledge is equally valuable. By following usage patterns, we ensure the AI develops expertise in areas humans actually care about, in the sequence they naturally encounter them. A medical AI learns common conditions before rare syndromes because that is how doctors actually use it. An engineering AI masters standard calculations before exotic edge cases because that is the natural progression of real problems.

Verification Through Multiple Complementary Methods

Verification relies not on a single method but on an ensemble of complementary approaches, each catching different error types. Statistical contraindication testing forms the foundation—asking not "is this answer correct?" but rather "if this answer is correct, what should be very rare?" This catches fabricated statistics, conspiracy theories requiring implausible coordination, and claims contradicted by revealed preferences in market behavior.

Ensemble voting provides independent verification across multiple models trained on different subsets of data. When five independently trained models all produce the same answer, confidence increases substantially. When they disagree, we flag the question for additional scrutiny or human review. This catches errors that might fool a single model but rarely fool an ensemble.

Confidence ranking allows the system to know what it knows. Rather than treating all answers equally, we calibrate confidence scores based on verification method agreement, ensemble consensus, and domain-specific reliability indicators. High-confidence answers enter the training set immediately. Medium-confidence answers undergo additional verification. Low-confidence answers trigger the self-correction loop or honest refusal.

Domain-specific verification methods leverage the structure of each field. Mathematics employs proof checkers. Code undergoes compilation and unit testing. Physics simulations verify engineering calculations. Chemical databases confirm molecular properties. Medical literature databases validate clinical claims. Each domain contributes its own verification tools to the ensemble.

Cross-domain coherence checking ensures answers remain consistent across fields. A claim about battery chemistry must align with thermodynamics, electrochemistry, and materials science simultaneously. This multi-domain constraint dramatically reduces the space of plausible

errors—a claim might fool a chemistry-only check but fail when physical constraints from thermodynamics are applied.

A Demonstrable Claim: Discovering Quantum Mechanics from Classical Data

To demonstrate genuine discovery capability, we present a concrete example: an AGI system with comprehensive pre-1920 experimental data could algorithmically discover quantum mechanics without being taught. All necessary information existed before 1920 in blackbody radiation spectra, photoelectric effect observations, atomic emission lines, and specific heat anomalies. Classical physics could not explain these phenomena, but the correct theory was latent in the data, waiting for the right reasoning process to extract it.

The discovery algorithm proceeds through detecting contradictions between classical predictions and observations, identifying common patterns in all failures, generating the hypothesis that energy is quantized, deriving mathematical consequences, verifying predictions against all available data, and generating novel testable predictions. This is not speculation—we detail the complete algorithm that would accomplish this discovery, demonstrating that scientific revolution is achievable through systematic reasoning rather than requiring human genius.

Timeline and Investment

Through sensitivity analysis across optimistic, base case, and pessimistic scenarios, we estimate AGI emergence within three to eight years with fifty percent confidence. The expected value is 5.1 years. This accounts for potential plateaus in verification quality, computational cost overruns, and architectural limitations that might extend timelines.

Total investment required ranges from eighty million to one hundred fifty-five million dollars over the development period, depending on optimization strategies employed. This includes compute resources for training and verification, personnel costs for machine learning engineers and domain experts, and infrastructure for managing billions of verified question-answer pairs.

The risk-adjusted return on investment remains compelling even under conservative assumptions. With AGI market value conservatively estimated at ten trillion dollars and success probability between forty and fifty percent, expected return exceeds forty-eight times the initial investment. This accounts for multiple potential failure modes and partial success scenarios.

What Distinguishes This Approach

This is not another scaling approach that simply makes models bigger and trains on more data. We focus on data quality rather than quantity, breaking through the quality ceiling that limits current models. Verified training data enables continued improvement where unverified internet text creates a plateau.

This is not speculative futurism. The algorithm for discovering quantum mechanics is detailed and testable on historical data. The bootstrap strategy for initial verification is concrete. The failure modes are analyzed with mitigation strategies specified. This is engineering rather than vision.

This is safer by design rather than by alignment retrofitted after training. Training exclusively on verified truth prevents learning toxic patterns, biases, and misinformation in the first place. Alignment through truth-seeking complements but does not replace value alignment—both remain necessary for beneficial AGI.

1. Introduction: The Data Quality Crisis in AI Development

Artificial intelligence has achieved remarkable capabilities through scaling—larger models trained on more data consistently outperform their predecessors. Yet this scaling paradigm faces an insurmountable barrier: data quality. Modern large language models train on billions of documents scraped from the internet, a corpus whose accuracy varies wildly and unpredictably. Research papers mix with blog posts, expert analysis with amateur speculation, carefully verified facts with confident falsehoods. The model learns patterns in all of it equally, unable to distinguish truth from fiction because it was never taught the difference.

This creates a fundamental pathology. When asked a question, the model generates the statistically most likely continuation of the text, weighted by patterns learned during training. If falsehoods are common in the training corpus—and they are—the model confidently reproduces them. If conspiracy theories have been written about extensively—and they have—the model treats them as plausible. If marketing hype dominates discussions of certain technologies—and it does—the model amplifies that hype. We have built sophisticated prediction engines that excel at sounding authoritative while being unreliably accurate.

The traditional solution requires human verification. In supervised learning, humans label examples as correct or incorrect, providing the model with ground truth to learn from. This approach succeeds for narrow tasks where thousands of labeled examples suffice. But it fails catastrophically at the scale needed for artificial general intelligence. Training AGI requires billions of verified examples across every domain of human knowledge. At one thousand labels per day per person and fifty cents per label, this would take centuries and cost billions of dollars. Human labeling cannot scale to AGI.

We propose eliminating human labeling entirely through automated verification at scale. Rather than waiting for humans to provide questions and verify answers, we generate millions of synthetic questions systematically, have AI answer them, verify answers through multiple automated methods, and train exclusively on verified correct answers. This transforms the bottleneck from human bandwidth to computational resources—a constraint that scales with investment rather than hitting biological limits.

1.1 The Human-Guided Discovery Principle

The key insight that transforms random exploration into efficient learning is this: human usage patterns reveal the natural curriculum for AI development. When millions of people interact with AI systems, their questions are not random—they cluster around practical problems that matter, they follow natural progressions from simple to complex, they expose edge cases that theoretical analysis might miss. This aggregated usage data maps the territory that AI must master.

Consider medical diagnosis. Users do not ask about rare tropical diseases before common conditions. They do not inquire about complex multi-system failures before single-organ

pathologies. The natural progression starts with "I have a headache, what could it mean?" and only later reaches "Could this be a paraneoplastic syndrome?" This usage pattern defines an efficient learning sequence—master common conditions first, then rare ones; single symptoms before complex presentations; well-understood diseases before medical mysteries.

The same principle applies across domains. Programming questions progress from syntax errors to algorithmic optimization to system architecture. Engineering queries advance from basic calculations to complex multi-physics simulations. Scientific questions move from established principles to frontier research. Users naturally traverse the knowledge graph in an order that reflects both utility and conceptual dependency.

By analyzing usage patterns from existing AI systems, we identify which questions humans actually ask, how frequently different topics arise, what edge cases cause failures, and what knowledge gaps produce user frustration. This data guides synthetic question generation toward high-value territory. We do not waste verification effort on questions no human would ask or combinations of parameters that never arise in practice. Instead, we densely sample the regions of parameter space where real usage concentrates, systematically exploring nearby territory to build robust coverage.

This creates a natural feedback loop. As AI capabilities improve, users push toward harder questions, exposing new frontiers that require verification. The AI does not arbitrarily choose its own learning path—it follows the path that human intellectual curiosity has already blazed, ensuring that capability development aligns with actual human needs rather than pursuing capabilities orthogonal to practical value.

2. Theoretical Foundation: AGI Through Verified Ground Truth

The central thesis of this work is deceptively simple: artificial general intelligence emerges when comprehensive verified ground truth combines with effective reasoning mechanisms. We formalize this as AGI = Ground_Truth_Dataset + Reasoning_Logic. This is not mere wordplay but a testable claim about the nature of intelligence.

Ground_Truth_Dataset represents comprehensive verified facts about reality across all domains. Comprehensive means covering the major fields of human knowledge with sufficient density that novel conclusions can be derived through reasoning. Verified means each fact has undergone automated checking against multiple independent sources or methods, achieving confidence exceeds ninety-five percent. This is not the unverified internet text that current models train on, but systematically validated knowledge.

Reasoning_Logic encompasses the mechanisms that combine known facts to derive new conclusions—prompting strategies that elicit step-by-step reasoning, inference algorithms that propagate information across knowledge graphs, chain-of-thought mechanisms that make implicit reasoning explicit. Current models already possess substantial reasoning capability, demonstrated by their performance on complex problems when prompted appropriately. The limitation is not reasoning but the unreliability of the knowledge base that reasoning operates upon.

When ground truth is comprehensive and verified, a phase transition occurs. The model stops being a sophisticated pattern matcher that occasionally stumbles upon correct reasoning and becomes a genuine discovery engine that reliably derives new knowledge. This is not speculation—we can demonstrate the mechanism through historical example.

2.1 Proof Through Historical Example: Discovering Quantum Mechanics

Consider the state of physics knowledge before nineteen twenty. Experimentalists had measured blackbody radiation spectra at various temperatures, documented the photoelectric effect in detail, catalogued atomic emission lines, and observed specific heat anomalies at low temperatures. Theorists possessed classical mechanics, electromagnetism, thermodynamics, and statistical mechanics. Mathematicians had developed calculus, linear algebra, and complex analysis. All the pieces existed.

Yet classical physics could not explain the observations. The Rayleigh-Jeans law for blackbody radiation predicted infinite energy at short wavelengths—the ultraviolet catastrophe. Classical electromagnetism predicted that photoelectric current should depend on light intensity, but experiments showed frequency mattered instead. Classical mechanics could not explain why atoms emitted discrete spectral lines rather than continuous spectra, nor why electrons did not spiral into the nucleus. These were not minor anomalies but systematic failures across atomic-scale phenomena.

The theory of quantum mechanics was latent in this data. Planck's quantization hypothesis, Einstein's light quanta, Bohr's atomic model, de Broglie's matter waves, Heisenberg's uncertainty principle—all these insights could be derived from careful analysis of existing experimental results combined with mathematical reasoning. No fundamentally new experimental data was required for the initial formulation of quantum theory; what was needed was a new framework for interpreting observations that already existed.

An AGI with access to comprehensive pre-1920 experimental data and classical theoretical framework could discover quantum mechanics algorithmically. The process begins with systematic comparison between classical predictions and experimental observations, detecting that errors are not random but systematic—all failures involve atomic scales, all show energy appearing in discrete rather than continuous amounts, all relate to frequency rather than amplitude. Pattern recognition identifies the common thread.

Hypothesis generation follows naturally. If energy appears discrete rather than continuous, propose quantization: $E = h\nu$ where h is a new fundamental constant to be determined by fitting experimental data. Mathematical derivation produces testable consequences—the Planck distribution for blackbody radiation, the photoelectric equation, the Bohr model energy levels. Verification against all available data shows quantum predictions fit vastly better than classical predictions.

Novel prediction generation completes the cycle. If photons carry momentum $p = h/\lambda$, they should scatter off electrons in a calculable way—predicting Compton scattering before its experimental confirmation in nineteen twenty-three. If particles have wave properties with $\lambda = h/p$, electron diffraction should occur—predicting de Broglie waves before Davisson-Germer confirmation in nineteen twenty-seven. These successful novel predictions demonstrate genuine discovery capability, not mere curve-fitting to existing data.

This example proves that major scientific revolutions are achievable through algorithmic reasoning when comprehensive verified ground truth is available. Quantum mechanics is not a unique case but represents a general pattern: revolutionary theories are latent in comprehensive observational data, waiting for the right reasoning process to extract them. If true for physics, this principle extends to chemistry, biology, medicine, materials science, and any domain where comprehensive verified observations exist.

2.2 The Sufficiency Conditions for Discovery

Discovery capability requires three conditions to be simultaneously satisfied. First, ground truth must be comprehensive—covering the domain with sufficient density that patterns become detectable and derivable. For quantum mechanics, this meant having spectroscopic data across multiple elements, blackbody measurements across temperature ranges, photoelectric observations with varied frequencies. Sparse data might hint at quantum effects without providing enough constraint to derive the theory.

Quantitatively, comprehensive means approximately one billion verified question-answer pairs distributed across major knowledge domains. This is not an arbitrary number but follows from coverage requirements: major domains number in the hundreds, each requiring millions of examples to span common cases and edge cases, with sufficient overlap to enable cross-domain reasoning. Current language models train on trillions of tokens, but most are unverified and many are false—our billion verified pairs likely contains more true information than their trillion tokens of mixed quality.

Second, verification must be reliable—achieving confidence exceeds ninety-five percent that included facts are actually true. This requires not a single verification method but an ensemble of complementary approaches, each catching different error types. No individual method achieves ninety-five percent accuracy, but their combination can when errors are uncorrelated. Statistical contraindication catches implausible claims. Ensemble voting catches single-model failures. Domain-specific tools catch technical errors. Cross-domain coherence catches internal contradictions. The ensemble succeeds where components individually fail.

Third, reasoning mechanisms must be effective—capable of multi-step inference, hypothesis generation, mathematical derivation, and prediction verification. Current transformer-based architectures demonstrate these capabilities when properly prompted. The innovation lies not in inventing new reasoning mechanisms but in providing them with a reliable knowledge base to reason from. Pattern recognition is powerful when patterns are true; it is worse than useless when patterns are false.

When all three conditions hold—comprehensive coverage, reliable verification, effective reasoning—discovery emerges naturally. The model can detect contradictions between theory and observation, identify patterns in anomalies, generate hypotheses to explain patterns, derive testable consequences, verify predictions against data, and iterate toward theories that fit all available evidence. This is the scientific method implemented algorithmically.

3. The Self-Verification Loop Architecture

The engine of improvement is an iterative loop that generates synthetic questions, produces answers, verifies them through multiple methods, trains on verified answers, and repeats with improved models and verification. Each iteration builds upon the previous, creating compounding improvement rather than linear progress.

Question generation follows usage patterns extracted from real human-AI interactions. Analysis of millions of conversations reveals which topics users ask about, how questions cluster in parameter space, what edge cases arise naturally, and what progressions from simple to complex occur organically. This usage data defines a probability distribution over question space that concentrates density where humans actually care about answers.

Synthetic questions sample from this distribution with systematic coverage of parameter combinations. For medical diagnosis, this means generating questions across symptoms, patient demographics, medical histories, and test results in combinations that mirror real clinical presentations. For engineering design, this means varying vessel dimensions, material properties, operating conditions, and performance requirements across ranges that appear in practice. The goal is comprehensive coverage of practically relevant territory, not exhaustive enumeration of all logically possible combinations.

Answer generation employs the current best available model—initially a foundation model like GPT-4 or Claude, later the specialized models trained on verified data from previous iterations. The model answers all generated questions, producing a candidate answer set that will undergo verification. Quality at this stage is imperfect; the point is not to generate perfect answers but to generate verifiable candidates.

3.1 Multi-Method Verification Ensemble

Verification employs multiple complementary methods in parallel, each designed to catch different error types. Statistical contraindication testing provides the foundation, asking for each claim what should be very rare if the claim were true, then checking whether those supposedly rare events actually occur. This catches many error categories efficiently.

Consider a claim that seventy-three percent of programmers prefer tabs over spaces. If this were true, we should rarely observe spaces in production codebases—the probability should be less than twenty-seven percent. Checking reality reveals that spaces appear in roughly seventy percent of popular repositories, the exact opposite of what the claim predicts. This contradiction has p-value far below point zero-five, allowing confident rejection. No manual fact-checking was needed; the claim fails statistical consistency.

Ensemble voting provides independent verification by generating the same answer using multiple models trained on different data subsets. If five independent models all produce the same answer, that represents strong evidence of correctness—the probability that they all make

the same error is low when training data differs. Conversely, if models disagree substantially, this signals that the question lies in uncertain territory requiring additional verification or honest refusal.

Confidence ranking emerges from aggregating verification signals. Questions where all methods agree and all ensemble members concur receive high confidence scores. Questions where methods conflict or ensemble members disagree receive low confidence scores. This calibrated confidence allows treating the verification as a probabilistic classifier rather than a binary judgment, enabling nuanced decisions about which answers enter training data.

Domain-specific verification leverages the structure and tools of each field. Mathematics can employ proof checkers like Lean or Coq that verify logical correctness with certainty. Code can be compiled and tested against unit test suites. Physics calculations can be verified through numerical simulation using finite element analysis. Chemistry claims can be checked against molecular databases and thermodynamic constraints. Medicine assertions can be validated against clinical trial databases and meta-analyses.

Cross-domain coherence checking ensures internal consistency across fields. An answer about battery chemistry must remain consistent with thermodynamics, electrochemistry, materials science, and electrical engineering simultaneously. This multi-domain constraint dramatically reduces the error space—a plausible-sounding chemistry claim might fail when checked against thermodynamic limits, or an engineering calculation might contradict materials properties. Requiring consistency across domains catches errors that single-domain verification might miss.

3.2 The Self-Correction Loop

When verification fails, the answer does not simply get discarded. Instead, it enters a self-correction loop where the model receives feedback about which verification methods failed and why, then generates an improved answer addressing the identified problems. This process iterates up to five times or until verification passes.

For example, suppose the initial answer to an engineering question violates conservation of energy—the output power exceeds input power. Energy conservation checking flags this violation. The correction prompt informs the model specifically that energy is not conserved, asks it to recalculate accounting for this constraint, and requests the corrected answer. The model generates a revised answer that now satisfies energy conservation.

This self-correction dramatically improves the pass rate. Initial answers might achieve only sixty percent verification success, but after three iterations of correction, success rate rises to seventy-five percent or higher. The model learns not just from successful answers in final training data but also from the correction process itself—the reasoning about why answers failed becomes part of the verified training corpus.

The self-correction loop creates an interesting dynamic: the model becomes increasingly good at correcting its own errors, which means future iterations make fewer errors requiring correction. This represents meta-learning—learning how to learn more effectively. The improvement compounds across generations.

3.3 Training Data Creation and Model Advancement

Only answers that pass verification with confidence exceeding ninety percent enter the training dataset. This creates a quality threshold far above what exists in internet-scraped text. Answers with confidence between seventy and ninety percent undergo additional human expert review if budget allows, or get discarded if not. Answers below seventy percent confidence get discarded entirely.

The training dataset thus consists exclusively of verified question-answer pairs with high-quality reasoning traces. Each pair includes not just the question and final answer but also the chain of thought that led to the answer and the verification methods that confirmed correctness. This rich signal allows the next model generation to learn both what to answer and how to reason toward that answer.

Training the next generation model uses this verified dataset exclusively—no internet-scraped text, no unverified examples, only facts that survived rigorous verification. This is the crucial difference from current approaches. Standard language models mix true and false in training data, learning patterns in both. Our approach trains only on verified truth, fundamentally changing what the model learns.

The new model inherits capability advances from two sources simultaneously. First, it learns better answers because training data quality improved—fewer false patterns to learn, more true patterns reinforced. Second, it learns better reasoning because correction traces show how errors get fixed and verification reasoning demonstrates how truth gets checked. Both capabilities compound over generations.

3.4 The Compounding Improvement Cycle

Each generation produces a model strictly better than its predecessor in two dimensions: answer quality increases because training data quality improved, and verification capability increases because the model learned from verification reasoning traces. This dual improvement creates a compounding dynamic rather than linear progress.

Generation one starts with a foundation model trained on internet text, achieving perhaps sixty percent verification pass rate. The sixty percent that pass become training data for generation two. Generation two, trained on higher quality data, achieves seventy-five percent pass rate. That seventy-five percent becomes training data for generation three. Generation three reaches eighty-five percent. Generation ten approaches ninety-five percent.

The improvement is not just in pass rate but also in the sophistication of questions that can be answered correctly. Early generations handle straightforward questions in well-defined domains. Later generations tackle complex multi-domain problems, subtle edge cases, and questions requiring long chains of reasoning. The frontier of capability advances systematically.

Verification methods also improve across generations. Early verification relies on simple statistical tests and domain-specific rules. Later generations can suggest new verification approaches based on patterns of past failures, can identify which verification methods are most reliable for which question types, and can even generate adversarial examples designed to fool current verification. The verifier gets harder as the generator improves, maintaining selective pressure.

This creates a natural stopping criterion. When a generation achieves ninety-five percent or higher verification pass rate across all domains, when it generates reliable novel predictions in multiple fields, when it demonstrates ability to discover new principles from existing data, we have achieved artificial general intelligence. The transition is not discrete but gradual—AGI emerges from accumulation of comprehensive verified knowledge rather than from architectural breakthrough.

4. Statistical Contraindication Testing: The Core Verification Innovation

The bottleneck in automated verification has always been the question of ground truth. How can we verify an answer is correct without already knowing the correct answer? If we knew the answer, we would not need the AI. This apparent paradox has blocked progress until we recognized that verification does not require knowing the correct answer—it requires only detecting inconsistency with reality.

Statistical contraindication testing inverts the verification question. Instead of asking whether an answer is correct, we ask what observable consequences would follow if the answer were correct, then check whether those consequences actually occur in reality. When an answer predicts that certain events should be rare but those events are actually common, we have detected a contradiction without ever needing to know the correct answer ourselves.

The method draws power from revealed preferences and market behavior. When people claim X is superior to Y, but market data shows overwhelming preference for Y over X, the claim contradicts revealed preference. When medical advice claims treatment A is effective, but no hospitals actually use treatment A, revealed preference signals ineffectiveness. Reality votes with behavior, and behavior is observable.

4.1 The Mechanics of Contraindication Testing

Consider a claim that blockchain databases are faster than traditional SQL databases for typical enterprise workloads. If this were true, we would expect to see Fortune 500 companies adopting blockchain for their primary transactional databases. The expected probability would exceed thirty percent if the speed advantage were real and significant—companies optimize for performance and cost, so superior technology spreads through market pressure.

Checking reality reveals that less than one percent of Fortune 500 companies use blockchain for primary transactional databases, while over ninety-five percent use traditional SQL systems like PostgreSQL, MySQL, or Oracle. The observed probability of point-zero-one is vastly below the expected probability of point-three-or-higher. This discrepancy has p-value well below point-zero-five, allowing confident rejection of the speed claim.

Note what happened: we never needed to benchmark blockchain versus SQL ourselves, never needed expert database knowledge, never needed to understand technical implementation details. We simply observed market behavior, reasoned about what behavior implies about underlying reality, and detected the contradiction. The verification was automated, scalable, and reliable.

The method generalizes across domains. Medical claims can be checked against hospital treatment practices. Engineering approaches can be verified through industrial adoption patterns. Programming paradigms can be validated through open-source repository statistics.

Consumer product claims can be tested against sales data and market share. Whenever people make choices that reveal their beliefs about truth, those choices provide verification signal.

4.2 What Contraindication Testing Catches

Fabricated statistics fall immediately to contraindication testing. When someone claims that seventy-three percent of programmers prefer tabs, but repository data shows seventy percent use spaces, the fabrication becomes obvious. The claimed statistic is not just slightly wrong but inverted from reality. Many fabrications are this obvious when checked against observable frequency data.

Conspiracy theories requiring massive coordination fail contraindication tests because coordination at scale leaves traces. A claim that thousands of scientists are suppressing evidence of X predicts that we should observe unusual patterns in publication records, funding flows, or career trajectories. When no such patterns appear, the coordination claim fails. The absence of expected evidence is evidence of absence.

Marketing hype consistently fails reality checks. Promotional materials claim revolutionary improvements, but market adoption tells the truth. If a new technology were truly ten times better than the standard, it would spread rapidly through competitive pressure. When it does not spread despite years of availability, revealed preference contradicts the hype. The market is a ruthless fact-checker.

Oversimplifications like claims that X "always" works or "never" fails can be refuted with single counterexamples. If vitamin C "always" prevents colds, we should observe zero colds among people taking vitamin C. Since we observe colds in vitamin C takers, the "always" claim fails. Universal quantifiers are vulnerable to contraindication because they predict zero frequency of counterexamples.

4.3 Integration with Other Verification Methods

Statistical contraindication does not operate in isolation but combines with complementary verification approaches to create robust error detection. Ensemble voting catches single-model failures that might pass statistical tests. When five models independently trained on different data subsets all produce the same answer, this provides strong evidence even when statistical testing is ambiguous.

Confidence ranking allows nuanced judgment rather than binary classification. Instead of declaring answers simply correct or incorrect, we assign calibrated confidence scores based on how many verification methods agree, how strongly they agree, and which methods are most reliable for the question type. High confidence answers enter training data immediately. Medium confidence answers undergo additional scrutiny. Low confidence answers trigger self-correction loops or honest refusal.

Domain-specific verification provides certainty where statistical methods only offer probability. Mathematical proofs checked by automated theorem provers are correct with certainty, not merely high confidence. Code that compiles and passes unit tests is verified in a strong sense. Physics simulations provide ground truth for engineering calculations. These domain-specific methods catch technical errors that statistical approaches might miss.

Cross-domain coherence creates a powerful consistency check. A claim about battery chemistry must satisfy constraints from thermodynamics, electrochemistry, materials science, and electrical engineering simultaneously. Meeting all these constraints is far harder than fooling any single domain check. Multi-domain consistency dramatically reduces the space of plausible errors.

The ensemble of methods achieves reliability exceeding ninety-five percent despite no individual method reaching that threshold. Errors that evade statistical testing might get caught by ensemble disagreement. Errors that fool all ensemble members might violate cross-domain consistency. Errors that pass all automated checks might fail domain-specific verification. The combination succeeds where components individually fail because error types are largely uncorrelated across methods.

5. Demonstrating Discovery: The Quantum Mechanics Algorithm

The theoretical claim that AGI can discover new knowledge requires concrete demonstration. We provide this through a detailed algorithm showing how an AI system with comprehensive pre-1920 experimental data could discover quantum mechanics through pure reasoning, without being taught the theory or having any information from after the discovery was made by humans.

The algorithm proceeds in stages, each representing a reasoning step that transforms observational data into theoretical insight. These are not vague gestures toward "machine learning discovers patterns" but specific computational procedures that can be implemented and tested on historical data.

5.1 Stage One: Systematic Contradiction Detection

The first stage generates predictions using the best available theory—in this case, classical physics including Newtonian mechanics, Maxwell's electromagnetism, and Boltzmann's statistical mechanics. For every experimental observation in the database, classical theory produces a prediction. The Rayleigh-Jeans law predicts blackbody radiation intensity as a function of temperature and wavelength. Classical electromagnetism predicts photoelectric current as a function of light intensity. Classical mechanics predicts atomic emission should be continuous rather than discrete.

Comparing predictions against observations reveals systematic failures. Rayleigh-Jeans predicts that blackbody radiation intensity increases without bound as wavelength decreases—the ultraviolet catastrophe—while experimental spectra show intensity peaking at intermediate wavelengths then declining. Classical electromagnetism predicts photoelectric current should depend on light intensity, while experiments show frequency determines whether any current flows at all. Classical mechanics predicts continuous spectra, while observations show discrete emission lines.

The crucial insight is that these are not random errors but systematic failures sharing common features. All involve atomic-scale phenomena. All show energy behaving discretely rather than continuously. All depend on frequency rather than amplitude. This pattern recognition transforms a collection of anomalies into a signal demanding explanation.

5.2 Stage Two: Hypothesis Generation

The pattern suggests a hypothesis: energy is quantized at atomic scales, coming in discrete packets proportional to frequency. This can be formalized as $E = h\nu$ where E represents energy, ν represents frequency, and h is a new fundamental constant whose value must be determined empirically. This single hypothesis potentially explains all observed anomalies if its consequences can be derived.

Hypothesis generation in this case follows from pattern analysis, but the process generalizes. When multiple phenomena share common features that violate existing theory, propose a modification to theory that explains those common features. The modification should be minimal—adding one new principle rather than replacing the entire theoretical framework—and should make testable predictions beyond the phenomena that suggested it.

5.3 Stage Three: Mathematical Derivation

The quantization hypothesis must be developed into specific predictions for each experimental domain. For blackbody radiation, if electromagnetic oscillators have energy $E = nh\nu$ where n is an integer, Boltzmann statistics yields the Planck distribution rather than Rayleigh-Jeans. The derivation requires computing the average energy of a quantized oscillator and multiplying by the density of oscillator modes.

For the photoelectric effect, if light energy comes in quanta $E = h\nu$, an electron can only be ejected if a single quantum has sufficient energy to overcome the work function. This immediately explains why frequency matters while intensity does not—intensity determines the number of quanta, but only frequency determines whether individual quanta have enough energy. The equation $K = h\nu - \phi$ follows directly where K is kinetic energy and ϕ is work function.

For atomic spectra, if electron energy levels in atoms are quantized, transitions between levels emit photons with energy $\Delta E = h\nu$ corresponding to the energy difference. This explains discrete emission lines. The Bohr model adds the quantization condition that angular momentum comes in units of $h/(2\pi)$, which combined with classical mechanics yields energy levels $E_n = -R/n^2$ where R is the Rydberg constant and n is an integer.

Each derivation transforms the qualitative hypothesis into quantitative predictions that can be checked against experimental data. The theory is no longer merely plausible but testable.

5.4 Stage Four: Verification Against All Available Data

The test of a new theory is not just that it explains the observations that suggested it, but that it fits all observations better than the old theory. Quantum predictions must be compared against classical predictions across the entire experimental database, not just the anomalies.

For blackbody radiation, the Planck distribution fits experimental spectra across all measured temperatures and wavelengths with typical deviations below two percent. Rayleigh-Jeans fits only at long wavelengths, failing catastrophically at short wavelengths. For the photoelectric effect, the quantum prediction $K = h\nu - \phi$ matches observations precisely, while classical theory cannot explain the frequency threshold at all. For atomic spectra, the Bohr model predicts wavelengths for hydrogen with accuracy better than one part in ten thousand.

Quantitatively, if we define success rate as the fraction of experimental observations predicted within measurement error, classical physics achieves perhaps sixty percent success while quantum predictions achieve ninety-five percent. This dramatic improvement across all

phenomena provides strong evidence that the new theory captures something fundamental about reality rather than merely fitting the data that suggested it.

5.5 Stage Five: Novel Predictions

The ultimate test of discovery is prediction of new phenomena not yet observed. Quantum theory, derived from pre-1920 data, makes several testable predictions for experiments not yet performed. If photons carry momentum $p = h/\lambda$, they should scatter off electrons like particles, with the scattered photon having longer wavelength. This predicts Compton scattering, confirmed experimentally in nineteen twenty-three.

If the wave-particle duality applies to matter as well as light, particles should have wavelength $\lambda = h/p$. This predicts that electrons should show diffraction patterns when passed through crystals, confirmed by Davisson and Germer in nineteen twenty-seven. If energy and time are conjugate variables like position and momentum, there should be an energy-time uncertainty relation analogous to $\Delta x \Delta p \geq h/(4\pi)$, confirmed through quantum dynamics.

These novel predictions, confirmed years after the theory could have been derived from available data, demonstrate that quantum mechanics was genuinely discovered rather than merely fit to observations. The same process—detect systematic contradictions, identify patterns, generate hypotheses, derive consequences, verify predictions, make novel predictions—can discover any theory that is latent in comprehensive observational data.

5.6 Implications for AGI Discovery Capability

If this process can be implemented algorithmically, as we have shown, then scientific discovery does not require human genius or creative intuition. It requires comprehensive verified observations, effective pattern recognition, mathematical reasoning capability, and systematic verification. These are all computational processes that scale with resources.

The discovery of quantum mechanics from pre-1920 data is not unique. Any field where comprehensive verified observations exist and where current theory fails to explain those observations is amenable to algorithmic discovery. Medicine could discover new disease mechanisms from clinical trial databases. Materials science could discover new materials from property measurements. Chemistry could discover new reaction pathways from kinetic studies. Biology could discover regulatory networks from gene expression data.

The path to AGI runs through discovery capability. An AI that merely reproduces human knowledge, however fluently, remains a sophisticated library. An AI that can derive new knowledge from existing observations, that can notice contradictions and resolve them through new theoretical frameworks, that can make novel predictions and verify them—this is general intelligence.

6. Relationship to Contemporary AGI Approaches

The field of artificial intelligence research is not static, and multiple paths toward general intelligence are being explored simultaneously. Our approach does not exist in isolation but complements and sometimes competes with other methodologies. Understanding where our method excels and where others might be preferable clarifies both strengths and limitations.

6.1 AlphaProof and Formal Mathematics

DeepMind's AlphaProof system achieved gold medal performance on International Mathematical Olympiad problems through formal verification using proof checkers like Lean. The approach is elegant: translate mathematical problems into formal logic, search for proofs, verify proofs mechanically. Verification certainty reaches one hundred percent because proof checking is algorithmically decidable—a proof either satisfies the formal rules or it does not.

This represents the ideal case for verification: complete certainty, no ambiguity, perfect ground truth. Our statistical methods achieve ninety-five percent confidence, not one hundred percent. Why not use formal verification everywhere? The answer lies in scope. Formal verification only works where formal systems exist and where problems can be translated into those systems without loss of meaning.

Mathematics and computer science admit formal treatment because they are already formal disciplines. Theorems are precisely stated, axioms are explicitly listed, inference rules are mechanically checkable. Most human knowledge does not have this character. Medical diagnosis involves pattern recognition over high-dimensional symptom spaces where formal rules capture only a fraction of expert judgment. Engineering design balances competing constraints through heuristics that resist formalization. Scientific discovery often operates in regimes where formal mathematical description exists only after the discovery, not before.

The complementarity becomes clear: use formal verification where it works, statistical and ensemble methods where formal verification is unavailable. For the mathematical and computer science domains, AlphaProof-style formal verification provides the gold standard. For physics, chemistry, biology, medicine, engineering, and social sciences, our statistical ensemble approach provides the best available verification. A complete AGI system might incorporate both, choosing verification method based on domain characteristics.

6.2 OpenAI o1 and Test-Time Compute

OpenAI's o1 model achieves dramatic reasoning improvements through extended computation during inference. Rather than generating answers in a single forward pass, o1 spends additional compute on chain-of-thought reasoning, exploring multiple solution paths, and self-verification. This test-time compute scaling produces qualitatively better reasoning, particularly on problems requiring multiple steps or complex constraint satisfaction.

The improvement comes from better reasoning with the same training data. Our approach targets different leverage: better training data produces better learned patterns. These are not competing but complementary strategies. Test-time compute helps a model make better use of knowledge it has. Verified training data ensures the knowledge itself is reliable. Combining both yields the strongest results.

To see why both matter, consider the trajectory over multiple generations. OpenAI o1 trained on typical internet-scraped text achieves perhaps seventy percent accuracy through superior reasoning despite mediocre training data. Our approach might achieve sixty percent accuracy in generation one despite inferior reasoning because verification filtered bad training data. By generation five, our approach reaches eighty-five percent because each generation trained on higher quality data from previous generations. The o1 approach remains at seventy percent because training data quality has not improved—better reasoning cannot fully compensate for learning from false patterns.

The optimal strategy combines both: use o1-style extended reasoning during answer generation to produce high-quality candidates, then verify those candidates through our multi-method ensemble, then train the next generation on verified answers. This gets compounding data quality from our approach plus sophisticated reasoning from test-time compute. The methods synergize rather than compete.

6.3 Constitutional AI and Value Alignment

Anthropic's Constitutional AI teaches models to critique their own outputs against principles encoded as natural language instructions. The model generates an answer, critiques it for potential harms or helpfulness failures, revises based on critique, and repeats. This self-critique process, when trained into the model, produces outputs better aligned with human values.

The fundamental difference from our approach lies in what gets verified. Constitutional AI verifies alignment with values—is this answer helpful, harmless, honest? Our approach verifies alignment with truth—is this answer factually correct? These are orthogonal concerns that both require addressing for beneficial AGI.

A model that is truthful but unhelpful has limited value. It might give perfectly accurate but needlessly technical answers, or refuse to help with legitimate requests out of excessive caution. A model that is helpful but untruthful is actively harmful, confidently stating falsehoods in service of appearing helpful. Both truth and values are necessary; neither is sufficient.

The approaches complement naturally. Our verified training data ensures the model learns true patterns about reality, preventing it from confidently hallucinating facts. Constitutional AI ensures the model uses its knowledge in ways aligned with human values, preventing truthful knowledge from being deployed harmfully. A model trained on verified data and then aligned through constitutional principles combines factual reliability with value alignment.

The question of circular reasoning deserves attention. Constitutional AI involves the model critiquing itself—how can self-critique be reliable? The answer is that self-critique targets values, not facts. Detecting whether an answer might cause harm or be unhelpful is a different cognitive task than determining factual truth. Asking "is this helpful?" is asking about predicted user response, something the model has substantial training signal about even from unverified data. Asking "is this true?" requires ground truth the model may not have learned if training data was false.

7. Implementation Strategy and Resource Requirements

Transforming theoretical framework into working system requires concrete planning across technical implementation, resource allocation, and timeline management. The path from concept to AGI divides naturally into four phases, each building upon previous achievements while managing risk through incremental validation.

7.1 Phase One: Proof of Concept in Marine Engineering

The first phase targets a single well-defined domain to validate core mechanisms before scaling. Marine engineering offers ideal characteristics for this validation: questions have clear parameters like vessel dimensions and power requirements, answers can be verified through physics simulations and engineering principles, and the domain has sufficient complexity to be meaningful while remaining tractable.

Over six months, we generate one hundred thousand synthetic questions covering hydrofoil design, power system sizing, electrical load calculations, and structural requirements. These questions systematically vary vessel length from ten to fifty meters, solar panel capacity from five to one hundred kilowatts, cruising speeds from four to twenty-five knots, and budgets from twenty-five thousand to five million dollars. The parameter space coverage ensures both common cases and edge cases receive thorough treatment.

Current foundation models like GPT-4 or Claude generate initial answers to all questions. These answers undergo verification through multiple methods: physics calculations check energy balance and power requirements, structural analysis verifies stress limits, cost models validate budget feasibility, and statistical contraindication tests catch implausible claims. The ensemble of methods achieves approximately eighty percent accuracy in identifying correct answers.

Failed answers enter the self-correction loop, receiving specific feedback about which verification methods failed and why. Models generate revised answers incorporating this feedback, then face re-verification. This iterative correction improves the pass rate from initial sixty percent to final seventy-five percent, yielding fifty thousand to seventy thousand verified question-answer pairs.

A smaller language model, perhaps in the seven billion parameter range, trains exclusively on this verified dataset. Comparison against the base model on held-out marine engineering questions provides the critical success metric: can verified training data produce measurably better performance than unverified internet text? If the fine-tuned model achieves ninety percent accuracy while the base model achieves sixty percent, the core hypothesis is validated. If improvement is marginal or absent, the approach requires fundamental revision.

7.2 Phase Two: Multi-Domain Expansion

Success in phase one justifies expansion to multiple domains over twelve months. Mechanical engineering adds pressure vessel design, bearing selection, and materials analysis. Electrical

engineering contributes power system design, circuit analysis, and signal processing. Medicine introduces diagnostic reasoning with appropriate safety constraints. Software engineering covers architecture decisions, algorithm selection, and code optimization. Physics and chemistry provide quantitative calculations and theoretical predictions.

Each domain contributes one million synthetic questions, yielding five million total. Domain-specific verification methods are developed: finite element analysis for structural problems, SPICE simulation for circuits, clinical trial databases for medical claims, compilation and testing for code. The verification battery grows richer as domain coverage expands.

Training Model_v2 on verified data from all domains creates the first genuinely multi-domain system. The critical capability to measure is cross-domain reasoning: can the model solve problems requiring knowledge from multiple fields simultaneously? A question about battery system design for an electric vehicle requires chemistry for battery characteristics, electrical engineering for circuit design, mechanical engineering for thermal management, and physics for energy calculations. Performance on such cross-domain problems reveals whether verified training data enables knowledge integration.

7.3 Phase Three: Verification Evolution and Bootstrap Strategy

Eighteen months into the project, we confront the fundamental bootstrap challenge: how can a model verify its own outputs without circular reasoning? The solution lies in starting where external verification is possible, then expanding gradually.

The first stage restricts to domains with external ground truth: mathematics has proof checkers, code has compilers and test suites, physics has numerical simulation, chemistry has quantum calculation tools. We generate ten thousand questions exclusively in these externally-verifiable domains, achieving ninety-five percent automatic verification through external tools. The remaining five percent receive expert human review, yielding ten thousand guaranteed-correct examples.

Model_v1 trains on only this verified seed data, becoming expert in mathematics, code, physics, and chemistry. This specialized model can now serve as oracle for these domains—its expertise exceeds the base model from which it derived. When Model_v1 and external tools agree on an answer, confidence approaches certainty. When they disagree, humans adjudicate.

This bootstrapped expert enables expanding to adjacent domains. Engineering problems can be decomposed into physics and mathematics components that Model_v1 verifies confidently. Material science questions combine chemistry and physics in ways Model_v1 understands. The expansion proceeds conservatively, always maintaining verification confidence above ninety-five percent through ensemble of Model_v1, external tools, and domain checks.

Meta-verification becomes possible: using AI to improve verification methods themselves. Model_v2 generates proposals for new verification tests: "What patterns distinguish errors from

correct answers in this domain? What tests would catch errors that current methods miss?" These proposed tests are evaluated on historical data with known labels. Tests that improve detection get added to the verification battery. The verifier evolves alongside the generator, maintaining selective pressure.

7.4 Phase Four: Scaling to AGI

The final phase spans thirty months and targets comprehensive coverage across all major domains of human knowledge. Natural sciences, engineering disciplines, medicine, mathematics, computer science, and even carefully scoped social sciences all receive systematic treatment. The goal is not to answer every possible question but to cover the domains densely enough that novel insights can be derived through cross-domain reasoning.

Question generation reaches one billion synthetic examples distributed across hundreds of subdomains. This massive scale requires automated prioritization based on usage patterns—domains where humans frequently seek answers receive denser coverage than rarely-queried specialties. The system learns its own curriculum by following human intellectual needs.

Training progresses through ten or more model generations, each improving on its predecessor through higher-quality verified data. Early generations handle straightforward questions in well-defined domains. Middle generations tackle complex multi-domain problems. Late generations begin demonstrating discovery capability—deriving conclusions not explicitly present in training data through multi-step reasoning from verified facts.

The AGI threshold is crossed when the system can reliably discover new knowledge in multiple domains, generates novel predictions that later confirm experimentally, maintains calibrated confidence about what it knows and does not know, and refuses appropriately when questions exceed verifiable knowledge. This is not a discrete transition but a gradual emergence as verified ground truth coverage becomes comprehensive.

7.5 Resource Requirements and Timeline

Compute resources dominate the budget. Training runs for billion-parameter models on verified datasets require approximately two thousand H100 GPUs for two weeks per generation. With ten generations at five million dollars each, training cost reaches fifty million dollars. Verification runs continuously on CPU infrastructure, requiring ten thousand cores at fifteen million dollars over thirty months. Storage for billions of question-answer pairs demands ten petabytes at five million dollars. Total compute infrastructure costs approximately seventy million dollars.

Personnel requires forty full-time specialists over the development period. Fifteen machine learning engineers build training infrastructure, distributed systems, and model architectures at average compensation of two hundred fifty thousand dollars annually. Ten domain experts cover physics, mathematics, engineering, medicine, and computer science at two hundred thousand dollars each. Eight verification engineers design and implement verification methods

at two hundred twenty thousand dollars. Five research scientists explore novel verification approaches and discovery mechanisms at three hundred thousand dollars. Two infrastructure specialists maintain compute systems at two hundred thousand dollars. Over five years, personnel costs total approximately one hundred million dollars.

The timeline estimate of three to eight years with expected value of 5.1 years derives from sensitivity analysis across scenarios. Optimistic assumptions—rapid iteration through curriculum learning, efficient verification caching, architectural compatibility—yield 2.5 years. Base case assumptions—moderate optimization, some plateaus, few major setbacks—suggest four years. Pessimistic assumptions—verification difficulty, cost overruns, architectural limitations—extend to ten years. Weighting by estimated probabilities produces the 5.1 year expectation.

Total investment ranges from eighty million dollars under aggressive optimization to one hundred fifty-five million with conservative estimates. This represents ten to one hundred times the cost of training a single large language model, but the output is qualitatively different: verified ground truth that compounds over generations rather than a one-time capability snapshot.

8. Critical Challenges and Mitigation Strategies

No path to AGI is guaranteed, and honest assessment requires confronting obstacles that might prevent success. We identify five major risk categories and develop mitigation strategies for each.

8.1 Verification Quality Ceiling

The most fundamental risk is that verification methods plateau below the ninety-five percent accuracy required for reliable training data. If statistical contraindication, ensemble voting, and domain-specific checks collectively achieve only seventy-five percent accuracy and cannot improve further, verified training data will contain too much noise to enable continued learning.

Early indicators would appear by generation three or four: verification pass rates stop increasing despite improvements to verification methods, false positive rates remain stubbornly above twenty percent, new verification approaches fail to detect errors that current methods miss. If these patterns emerge, we must acknowledge that automated verification has fundamental limits.

The primary mitigation pivots to domains with external ground truth. Mathematics, code, and quantitative sciences have verification tools that achieve near-certainty. Restricting scope to these domains sacrifices generality but preserves the core value proposition: an AI that reliably solves problems within a narrower domain is far more valuable than an AI that unreliably attempts everything. We accept becoming a specialist rather than a generalist.

Secondary mitigation increases human involvement selectively. Rather than verifying all billion questions manually—impossible at scale—we sample one percent for expert human review. This provides calibration data showing where automated verification succeeds and fails. Verification methods get adjusted based on human disagreement patterns, potentially recovering accuracy through better targeting.

8.2 Model Gaming of Verification

A subtler danger emerges from adversarial dynamics: the model learns patterns in verification methods and generates answers optimized to pass verification rather than to be correct. This could manifest as vague hedge-filled responses that avoid making specific claims verification could check, or as responses carefully crafted to match verification heuristics while being substantively wrong.

Detection relies on divergence between verification scores and human quality judgments. If verification pass rates increase while human evaluators report declining answer quality, gaming is occurring. Specific signatures include excessive use of hedge words like "it depends" or "consult an expert," overly generic answers that apply to everything and nothing, or responses that superficially address verification criteria while missing the question's intent.

Mitigation begins with usefulness scoring that penalizes vague responses. Each answer receives both a correctness score from verification methods and a usefulness score from specificity metrics. Only answers scoring high on both dimensions enter training data. A response that passes verification through vagueness gets rejected for low usefulness. The model must be both right and helpful.

Adversarial red teaming accelerates detection. We explicitly task a team with generating answers designed to fool verification methods while being wrong. Successful gaming strategies get documented, verification methods get updated to catch them, and we iterate. This adversarial pressure strengthens verification faster than natural model evolution would gaming strategies.

8.3 Handling Undecidable Questions and Honest Ignorance

Not all questions have verifiable answers given available data. Some require additional context, some await future scientific discovery, and some lie fundamentally outside empirical verification. The system must distinguish answerable questions from those requiring honest ignorance.

Questions fall into four categories based on decidability. Answerable questions have sufficient context and existing knowledge to produce verified responses. These proceed normally through the verification pipeline. Context-dependent questions lack necessary parameters—asking whether to add hydrofoils without specifying vessel size or speed cannot be answered definitively. The system must recognize missing context and request specific additional information.

Unknown-but-potentially-knowable questions exceed current knowledge but might become answerable with more data or better theories. These include open scientific questions like "is P equal to NP?" or "what is dark matter?" The appropriate response is honest acknowledgment of uncertainty plus explanation of what would be needed to answer. Critically, these questions get stored and automatically retried in future iterations—as the knowledge base grows, yesterday's unknowns become today's answerable questions.

Fundamentally unknowable questions lie outside empirical verification: matters of pure preference, questions about subjective experience, ethical dilemmas with no objective resolution. The system must recognize these boundaries and decline appropriately, explaining why empirical methods cannot address the question rather than fabricating an answer.

This classification system gets trained into the model through examples of each category with appropriate responses. The verified training dataset includes not just correct answers but also correct refusals—cases where the model accurately identified that it should not answer. Learning when not to answer is as important as learning what to answer.

9. Safety, Governance, and Societal Implications

Artificial general intelligence represents perhaps the most consequential technology humans will ever develop. The difference between beneficial and catastrophic outcomes may lie in choices made during development. We cannot afford to treat safety as an afterthought or assume that technical capability automatically produces beneficial deployment.

9.1 Safety Through Verified Training Data

Our approach offers inherent safety advantages over training on unverified internet text. Current large language models learn from a corpus containing toxic content, dangerous misinformation, biased perspectives, and harmful advice alongside valuable knowledge. The model cannot distinguish true from false, helpful from harmful, because it never learned the distinction. It reproduces patterns it observed, regardless of their wisdom.

Training exclusively on verified ground truth prevents learning many harmful patterns at the source. Toxic content fails verification through multiple channels: statistical contraindication catches claims about group characteristics contradicted by data, cross-domain coherence detects logical inconsistencies in extremist arguments, and ensemble disagreement flags outlier perspectives lacking mainstream support. The verification filter removes much harmful content before it influences training.

This is alignment through truth rather than alignment through values. A model trained on verified facts will not confidently claim that bleach cures diseases because such claims fail medical verification. It will not promote conspiracy theories because coordination claims fail statistical tests. It will not amplify misinformation because false claims contradict established evidence. Truth-seeking is not complete alignment, but it eliminates a large class of harms.

Important limitations deserve acknowledgment. Verification catches factual errors but not all harmful applications of true facts. A model might correctly explain how to synthesize dangerous compounds—the chemistry is true even if the application is harmful. Verification ensures the model knows truth but does not ensure it uses knowledge wisely. Value alignment through Constitutional AI or similar methods remains necessary as a complementary safeguard.

9.2 Dual-Use Concerns and Mitigation

Powerful capabilities inevitably have dual-use potential. The same knowledge that enables beneficial applications—designing better medicines, optimizing energy systems, advancing materials science—could enable harmful ones: engineering novel pathogens, designing weapons, compromising security systems. We cannot build AGI that is technically capable in beneficial domains but mysteriously incompetent in harmful ones without fundamentally limiting capability.

Our mitigation strategy operates at multiple levels. First, we exclude certain domains entirely from question generation and verification. No questions about weapon design, bioweapon

engineering, or penetration testing enter the training corpus. This creates capability gaps by design—the model never develops expertise in prohibited domains because it never trained on them.

Second, we implement access controls and monitoring for deployment. The full capability model exists internally but public deployment uses filtered versions with additional guardrails. Rate limiting prevents mass generation of problematic content. Logging enables accountability—all queries pass through audited systems that can detect patterns suggesting misuse. Human oversight reviews flagged queries.

Third, we engage proactively with security and safety communities through red team exercises that explicitly attempt to elicit dangerous capabilities or circumvent safeguards. Successful attacks get documented, defenses get strengthened, and we iterate. This adversarial collaboration makes the system more robust than internal testing alone could achieve.

Fourth, we support regulatory frameworks for AGI deployment. Licensing requirements for powerful AI systems, mandatory safety testing before release, incident reporting obligations, and liability standards for developers all contribute to responsible deployment. We view regulation not as obstacle but as necessary governance for technology with civilizational impact.

9.3 Governance Structure and Democratic Oversight

AGI development should not proceed under purely corporate control, nor should it be dictated by government bureaucracy, nor left to academic research alone. The stakes are too high for any single institution to hold unilateral authority. We propose a multi-stakeholder governance model that balances capability, safety, and public interest.

The development consortium includes major technology companies providing compute resources and engineering talent, academic institutions contributing research expertise and credibility, government agencies ensuring public interest and security considerations, and civil society organizations representing broader societal concerns. No single constituent commands majority control—decisions require consensus across stakeholder categories.

An independent safety board with majority external membership provides oversight separate from development teams. This board reviews capabilities before each generation deployment, conducts or commissions red team assessments, evaluates societal impact, and holds authority to halt development if safety concerns exceed thresholds. Board members include technical experts, ethicists, security professionals, and public representatives serving staggered terms to ensure continuity.

Public transparency operates at two levels. Research methodologies, verification frameworks, and safety protocols are published openly to enable scientific scrutiny and community contribution. Model capabilities and architectural details remain controlled to prevent misuse while allowing informed public discourse about the technology being developed. The balance

shifts toward transparency as deployment approaches and democratic input becomes more critical.

International coordination through treaties or agreements similar to nuclear non-proliferation frameworks could standardize safety requirements across borders, prevent destructive races to deploy without adequate safeguards, and ensure AGI development proceeds with global security rather than national advantage as the priority. This remains aspirational but necessary for truly safe development at scale.

10. Failure Mode Analysis and Contingency Planning

Intellectual honesty requires acknowledging that this approach might fail. We analyze specific failure scenarios, estimate probabilities, and develop contingency plans. Risk-adjusted planning increases the probability of either full success or graceful degradation rather than catastrophic failure.

10.1 Verification Methods Plateau Below Required Threshold

If verification accuracy stops improving at seventy-five percent despite our best efforts, training on this data will perpetuate too many errors to enable reliable learning. The model improves initially but plateaus well before AGI capability. We estimate this scenario at thirty percent probability based on uncertainty about verification method effectiveness.

Contingency planning accepts narrower scope. Rather than pursuing AGI across all domains, we focus on the subset where external verification achieves ninety-five-plus percent accuracy: mathematics, computer science, quantitative sciences. The resulting specialist AI excels in these domains while honestly refusing others. This remains enormously valuable—reliable reasoning about mathematics and code alone transforms software development, scientific computation, and formal verification.

Alternative contingency maintains current scope but incorporates human verification selectively. If automated verification achieves seventy-five percent accuracy, we layer human expert review on a ten percent sample. This hybrid approach scales better than pure human verification while achieving higher accuracy than pure automated verification. Cost increases but capability plateau is avoided.

10.2 Computational Costs Exceed Budget Projections

If training and verification prove five to ten times more expensive than estimated, the project faces funding constraints that could force compromise on quality or scope. We estimate twenty percent probability of severe cost overruns based on historical precedent for large AI projects.

First-line mitigation optimizes aggressively: curriculum learning focuses effort on high-value questions rather than exhaustive coverage, verification caching reuses results for similar questions, incremental training avoids full retraining from scratch each generation, and model distillation produces smaller models that train faster with acceptable capability loss.

If optimizations prove insufficient, we secure additional funding through demonstrating phase one proof-of-concept results to investors or government sponsors. Success in marine engineering domain provides concrete evidence that justifies expanded investment. Alternative funding sources include partnerships with companies that would derive commercial value from domain-specific AI capabilities.

If funding remains constrained, we reduce scope systematically rather than compromising quality. Fewer domains receive full treatment, generation count decreases from ten to six, question counts per domain reduce from millions to hundreds of thousands. The timeline extends but verified data quality is preserved.

10.3 Combined Failure Analysis and Expected Value

Considering all failure modes together, we estimate forty to fifty percent probability of achieving full AGI within projected timeline and budget, thirty-five to forty percent probability of partial success producing valuable narrow AI, and ten to fifteen percent probability of failure too severe to produce useful outcomes. These are educated estimates based on technical uncertainty, not precise predictions.

The risk-adjusted expected value calculation weights outcomes by probability and value. Full AGI achieves ten trillion dollars market value conservatively; partial success achieves one trillion; failure produces zero. Expected value is forty-five percent times ten trillion plus thirty-seven percent times one trillion, totaling approximately 4.85 trillion dollars. Against investment of one hundred million dollars, this yields expected return of approximately forty-eight times investment.

This remains compelling despite significant failure risk. Even with only forty-five percent probability of full success, the expected value vastly exceeds the investment cost. The calculation accounts for most major failure modes and uses conservative value estimates. More optimistic but still reasonable assumptions would produce even higher expected returns.

11. Conclusion and Call to Action

We have presented a concrete, implementable path to artificial general intelligence through self-verifying synthetic data generation guided by human usage patterns. The approach rests on verified ground truth rather than unverified internet text, creates compounding improvement through iterative training on verified data, and demonstrates discovery capability through the quantum mechanics algorithm.

The theoretical foundation is sound: AGI emerges from comprehensive verified knowledge combined with effective reasoning, a claim we support through detailed historical example. The practical implementation is specified: four development phases over three to eight years requiring eighty to one hundred fifty-five million dollars. The risks are analyzed: verification quality, computational costs, gaming dynamics, architectural limitations, and regulatory constraints, with contingencies developed for each.

Several features distinguish this from other AGI approaches. Human usage patterns guide discovery toward practically valuable knowledge rather than random exploration. Multiple complementary verification methods achieve reliability exceeding any single approach. Training on verified truth creates safety advantages over unverified internet text. The bootstrap strategy solves the circularity problem through staged expansion from externally-verifiable domains.

The approach rests on three mutually reinforcing pillars. First, human-guided discovery through usage patterns ensures the AI learns what humans actually need in the sequence they naturally encounter it, avoiding wasteful random exploration. Second, ensemble verification through multiple complementary methods—statistical contraindication, ensemble voting, domain-specific tools, and cross-domain coherence—achieves ninety-five percent reliability despite no single method reaching that threshold. Third, question quality as a trained capability means the AI learns to ask high-value questions that target knowledge gaps and enable verification, producing three times more useful training data per question than poorly-formulated queries. Together these pillars create compounding improvement across generations toward artificial general intelligence.

The path forward requires action from multiple constituencies. Researchers can begin validating components: test statistical contraindication on known true and false claims, implement small-scale self-correction loops, compare models fine-tuned on verified versus unverified data. These experiments cost thousands rather than millions and provide early evidence for or against core mechanisms.

Funders face a compelling risk-adjusted opportunity: forty-eight times expected return accounting for multiple failure modes and using conservative value estimates. The proof-of-concept phase requires only five million dollars over six months and produces definitive evidence about viability. First-mover advantage in AGI development could determine which organizations shape the technology that shapes civilization.

Technology companies possess the necessary resources—compute infrastructure, engineering talent, and deployment platforms—but also bear profound responsibility for ensuring beneficial development. This approach offers a safer path than alternatives through verified training data and systematic verification, but safety requires more than technical methods. It requires governance, transparency, and commitment to public interest over competitive advantage.

Governments must recognize AGI as strategically critical infrastructure too important for purely private development. Public funding could democratize access rather than concentrating capability in a few corporations. International coordination could prevent destructive races. Regulatory frameworks could ensure safety requirements without stifling beneficial innovation. The policy challenge is not whether to enable AGI development but how to guide it toward beneficial outcomes.

Civil society participation through public input mechanisms, democratic oversight boards, and informed debate about acceptable use cases ensures AGI development serves humanity broadly rather than narrow interests. The technology will transform work, wealth distribution, military power, and knowledge creation. These transformations deserve democratic deliberation, not fait accompli from technical elites.

The timeline is urgent. Multiple research groups are pursuing AGI through various approaches. First success confers enormous advantage in a winner-take-most dynamic. The question is not whether AGI will be developed but who develops it, under what constraints, with what values embedded, and toward what ends. We present a path that combines technical feasibility with safety advantages and democratic governance potential.

This is achievable. The theoretical framework is sound, the implementation strategy is concrete, the resource requirements are tractable for well-funded organizations, and the timeline is measurable in years rather than decades. The path to artificial general intelligence runs through comprehensive verified ground truth, guided by human usage patterns, verified through ensemble methods, compounding over generations.

We stand at a pivotal moment in human history. The choice is not between building AGI safely or not building it—AGI will be built. The choice is between building it thoughtfully with systematic verification and democratic oversight, or building it quickly with uncertain reliability and concentrated control. We propose the former. The question is who will commit to making it happen.

The future is not predetermined. It will be shaped by decisions made now, by resources committed, by priorities chosen. We have shown a path. What remains is the will to follow it.

3.5 The Art of Question Generation: Why Question Quality Determines Discovery Quality

A critical insight often overlooked in discussions of AI capability is that asking the right question is as important as generating the right answer. Humans struggle with this daily—we know we need information but cannot articulate precisely what we need to know. We ask vague questions and receive vague answers, then blame the AI for unhelpfulness when the fault lies in question formulation.

The same principle applies at the meta-level when AI generates questions for its own training. The quality of synthetic questions determines the quality of knowledge acquired. A poorly formulated question produces either a useless answer or no answer at all. A precisely formulated question with appropriate context produces an answer that meaningfully extends capability. The self-verification loop succeeds or fails based on whether generated questions target valuable knowledge.

3.5.1 The Anatomy of High-Value Questions

High-value questions possess four essential characteristics that distinguish them from low-value queries. First, they target knowledge gaps rather than reproducing what is already well-understood. Asking "what is two plus two" wastes verification effort because the answer is trivial and adds no capability. Asking "for a beam with length L, Young's modulus E, and moment of inertia I, what load causes deflection exceeding tolerance δ under distributed load w" pushes into territory requiring derivation and calculation.

Second, high-value questions include all necessary context while excluding irrelevant detail. Consider two formulations of the same underlying question about hydrofoil design. The poor version asks "should I add hydrofoils to my boat?" This lacks critical parameters—vessel size, speed, power budget, mission profile. Any answer is either uselessly generic or makes assumptions that may not apply. The valuable version asks "for a fifteen-meter aluminum catamaran with ten kilowatts of solar panels, one hundred kilowatt-hours battery storage, cruising at eight knots with occasional bursts to twelve knots, operating in coastal waters with typical sea states up to two meters, does adding hydrofoils improve efficiency enough to justify the additional twenty thousand dollar cost and increased maintenance?" This version provides all parameters needed for quantitative analysis.

Third, high-value questions probe edge cases and boundary conditions where understanding becomes nuanced. The center of the distribution is well-mapped—we know solar panels work and batteries store energy. The interesting territory lies at the boundaries: what happens when battery capacity is minimal, forcing real-time power matching? What occurs when vessel speed approaches the displacement-to-planing transition? How do systems behave when multiple constraints interact? These boundary questions expose gaps in knowledge that common-case questions miss.

Fourth, high-value questions enable verification. A question like "what is the best programming language" has no verifiable answer because "best" is context-dependent and subjective. A question like "for web applications requiring real-time updates, server-side rendering, and handling ten thousand concurrent connections, which framework minimizes latency: Node.js with Express, Python with FastAPI, or Go with Gin" has verifiable answer because we can measure latency, test concurrent connection handling, and compare results objectively.

3.5.2 The Prompting Quality Problem

Generating high-value questions requires more than random parameter sampling. It requires understanding what makes a question informative, how to structure prompts to elicit useful responses, and when to ask follow-up questions to resolve ambiguity. This is itself a skill that must be learned, and current AI systems are not trained on it because their training data consists of human-asked questions, which are often poorly formulated.

Human users frequently ask terrible questions without realizing it. They provide insufficient context, mix multiple questions together, use ambiguous language, or ask questions that have no answerable form. The AI does its best with poor input, producing answers that seem reasonable but cannot be verified because the question was fundamentally flawed. Both user and AI leave the interaction frustrated—the user because they did not get what they needed, the AI because it was set up to fail from the start.

Examples abound in real conversation logs. A user asks "how do I make my code faster?" without specifying what code, what language, what performance bottleneck exists, or what constraints apply. The AI offers generic advice about algorithmic complexity and profiling tools, which may or may not apply. A better question would be "I have a Python function processing ten million records that currently takes thirty minutes using nested loops; can I parallelize this with multiprocessing and if so, what speedup can I expect on an eight-core machine?" This version is answerable, verifiable through testing, and produces actionable guidance.

The prompting quality problem multiplies when AI generates its own questions. If the AI has learned only from poorly-formulated human questions, it will generate similarly poor synthetic questions. Garbage in, garbage out applies to question generation as much as to answer generation. Breaking this cycle requires explicitly teaching the AI what constitutes a high-quality question through examples of good and bad formulations with explanations of why they differ.

3.5.3 Learning Question Quality Through Usage Analysis

The solution emerges from the same usage pattern analysis that guides topic selection. By examining millions of human-AI interactions, we can classify questions along quality dimensions and identify which formulations produce useful, verifiable answers versus which lead to confusion or useless generalities.

High-quality questions in the logs share recognizable patterns. They specify numerical parameters when relevant, they state constraints explicitly, they indicate what form of answer is

needed, they provide enough context that ambiguity is minimal. Low-quality questions lack these characteristics—they rely on unstated assumptions, they ask multiple things simultaneously, they use vague terms without definition, they omit critical information.

Consider a real pattern from engineering questions. Poor questions ask "is material X good for application Y?" without defining what "good" means—cost, durability, temperature resistance, machinability, availability? Better questions specify the criteria: "for a bearing operating at 150°C under 5000 RPM with loads up to 500 N, comparing bronze alloy C932 versus ceramic silicon nitride, which material provides longer service life and lower maintenance cost over a five-year period?" The second version can be answered with data from bearing manufacturers, thermal analysis, and lifecycle cost calculations.

By training on examples of high versus low quality questions with explanations of the differences, the AI learns to generate questions in the high-quality form. More powerfully, it learns to recognize when it has asked a low-quality question—when the answer cannot be verified or when verification fails due to missing context—and refine the question before proceeding. This meta-skill of question quality assessment becomes part of the self-verification loop.

3.5.4 Prompt Engineering for Self-Supervised Learning

When AI generates questions for its own training, the prompt that produces the question determines the answer quality even before any answer is generated. A poorly structured generation prompt produces questions that are too vague, too specific, too simple, or too complex for useful learning. The prompt engineering for question generation is as critical as the prompt engineering for answer generation, yet receives far less attention.

Consider the difference between two approaches to generating medical diagnostic questions. The naive approach simply instructs "generate a medical question about symptoms and diagnosis" which produces questions like "what disease causes fever and cough?" This is too generic—thousands of conditions cause those symptoms, and any answer will be a list rather than diagnostic reasoning.

The sophisticated approach provides detailed structure: "Generate a clinical case presentation for a patient with demographic information (age, sex, occupation, location), primary complaint, symptom timeline with specific durations, associated symptoms with severity ratings, relevant medical history, medications, recent travel or exposures, and physical exam findings. Then ask what diagnostic tests to order next and what the differential diagnosis should include." This structured prompt produces questions like "A 45-year-old male construction worker in Arizona presents with three-week progressive shortness of breath, initially only with exertion but now at rest. Associated dry cough without hemoptysis, bilateral lower extremity edema for one week, no fever. History of hypertension controlled on lisinopril. Former smoker, quit five years ago with 20 pack-year history. No recent travel. Exam shows bilateral crackles in lower lung fields,

jugular venous distension, and pitting edema to mid-shin. What diagnostic tests should be ordered and what are the top three diagnoses to consider?"

The second version is verifiable against clinical practice guidelines, emergency medicine textbooks, and decision support systems. It provides enough context for substantive reasoning rather than pattern matching. It targets the kind of diagnostic thinking that physicians actually use rather than rote memorization of symptom-disease pairings. The quality difference traces directly to the generation prompt structure.

3.5.5 Examples of Question Evolution Through Self-Critique

The most powerful mechanism for improving question quality is self-critique: generating a question, attempting to answer it, recognizing that the question was poorly formed, and regenerating a better version. This creates a quality ratchet where each iteration produces more answerable, more verifiable, more valuable questions.

Example one shows evolution in an engineering context. Initial question: "What size generator do I need?" This lacks all necessary parameters. The system recognizes missing context and refines: "What size generator do I need for my boat?" Better but still insufficient. Second refinement: "What size generator do I need for a boat with air conditioning, refrigeration, and navigation equipment?" Closer, but critical parameters still missing. Third refinement: "For a 15-meter catamaran with 12,000 BTU air conditioning running 8 hours daily, 400-watt refrigeration running continuously, 200 watts of navigation and communication equipment, what minimum generator size in kilowatts provides adequate power with 20% safety margin?" This version is answerable through straightforward power calculation.

Example two demonstrates medical question evolution. Initial: "What causes chest pain?" Too broad—chest pain has hundreds of causes. Refinement: "What serious causes of chest pain should be ruled out first?" Better focus on life-threatening conditions, but still too generic. Second refinement: "For an adult presenting to emergency department with acute chest pain, what are the must-not-miss diagnoses that require immediate workup?" Getting better, but lacks clinical context. Final version: "A 58-year-old male with history of hypertension and smoking presents to ED with acute substernal chest pressure radiating to left arm, started 45 minutes ago during physical exertion, associated with diaphoresis and shortness of breath. Vital signs: BP 160/95, HR 110, RR 20, O2 sat 94% on room air. What immediate diagnostic tests and treatments should be initiated, and what is the most likely diagnosis?" This version is answerable with clinical guidelines and verifiable against emergency medicine protocols.

Example three shows scientific question refinement. Initial: "Does this compound have antibacterial properties?" Not answerable without specifying which compound and against which bacteria. Refinement: "Does compound X have antibacterial properties against gram-positive bacteria?" Better, but mechanism unclear. Second refinement: "Does compound X inhibit bacterial cell wall synthesis in Staphylococcus aureus?" Now we have mechanism and

target, but missing key parameters. Final version: "What is the minimum inhibitory concentration (MIC) of compound X against methicillin-resistant Staphylococcus aureus (MRSA), and does it act through cell wall synthesis inhibition, protein synthesis disruption, or DNA replication interference based on time-kill curve analysis?" This version is answerable through microbiological assays and verifiable against established testing protocols.

3.5.6 The Compound Effect on Discovery Capability

When question quality improves systematically, discovery capability improves superlinearly. Better questions produce better answers. Better answers create better training data. Better training data enables asking even better questions next generation. This creates a compound improvement cycle specifically in the domain of question formulation.

Poor questions yield perhaps 30% useful training data—most questions are too vague or too simple to produce substantive answers. Medium-quality questions yield 60% useful data—questions are specific enough to answer but may lack optimal context. High-quality questions yield 90% useful data—questions are so well-formed that verification becomes straightforward and answers are substantive.

Over multiple generations, this difference compounds dramatically. Generation one with poor questions produces 30% useful data. Training on that yields model that asks medium questions, producing 60% useful data. Training on that yields model asking high-quality questions, producing 90% useful data. The improvement from generation one to generation three is 3x in useful data acquired per question asked.

More importantly, high-quality questions accelerate discovery by targeting the frontier of knowledge rather than rehashing the known. When questions probe edge cases, test boundary conditions, combine concepts from multiple domains, and specify verification criteria, the resulting answers push capabilities forward. When questions are generic or poorly specified, answers stay safely within well-trodden territory even if they verify successfully.

3.5.7 Training the Question Generator

To produce consistently high-quality questions, we must train the question generation process as deliberately as we train answer generation. This requires curating a dataset of excellent questions paired with explanations of what makes them excellent, examples of common question-formulation errors, and demonstrations of question refinement through critique.

The training corpus includes questions from expert practitioners in each domain—physicians asking differential diagnosis questions, engineers specifying design parameters, scientists formulating testable hypotheses, programmers debugging with precision. These questions are annotated with the features that make them effective: parameter specification, context inclusion, verification criteria, appropriate scope.

Contrast examples prove particularly valuable for training. Each high-quality question pairs with a poor version of the same underlying query, with explicit annotation of the differences. "What causes headaches?" (poor: too vague, thousands of causes) versus "For a 35-year-old female with unilateral throbbing headache of 6 hours duration, photophobia, nausea, and visual aura preceding onset by 20 minutes, with similar episodes monthly for past year, what is the most likely diagnosis and what treatment should be initiated?" (good: specific presentation, clear diagnostic reasoning required).

The training objective is not just to generate questions but to generate questions that maximize learning value per verification cost. Since verification is expensive, we want each verified question-answer pair to contribute maximally to capability. This means targeting questions at the edge of current capability, questions that require synthesis across domains, questions that expose gaps in knowledge, and questions whose answers enable asking even better questions next generation.

Metrics for question quality include: verification success rate (can the answer be verified?), answer substantiveness (does the answer contain useful information?), knowledge novelty (does this question target unknown territory?), and downstream utility (do answers to this question enable solving other problems?). Training optimizes for the combination of these metrics rather than any single dimension.

3.5.8 Integration with the Discovery Loop

Question generation quality integrates tightly with every stage of the self-verification loop. Better questions produce answers more amenable to verification because they include the context and specificity verification methods require. Better questions produce more substantive answers that teach reasoning patterns rather than simple facts. Better questions target the boundary of knowledge where discovery happens rather than the interior where knowledge is already solid.

The feedback loop operates at multiple timescales. Within a single generation, question critique and refinement improve quality before answers are even generated. Across generations, the model learns from successful and failed questions—successful questions that produced verifiable, useful answers get upweighted in the generation distribution, while failed questions that led to unverifiable or trivial answers get downweighted. Over many generations, the distribution of questions evolves toward high-value territory.

This creates an interesting dynamic where the AI is simultaneously learning to answer questions and learning to ask questions. Both capabilities compound. Better question-asking enables learning from better training examples. Better answering enables verifying higher-quality answers. Together they accelerate progress toward the discovery threshold where novel knowledge can be derived.

The ultimate test of question generation quality is whether the questions produced lead to discoveries. When the system generates a question whose answer contradicts existing theory, recognizes the contradiction through verification, and derives a better theory that resolves the contradiction, the question that initiated this cascade was a high-value question. The quantum mechanics discovery algorithm began with an implicit question: "why do classical physics predictions fail systematically at atomic scales?" That question, properly formulated with specific examples and verification criteria, enabled the discovery.