HLLM: Hamiltonian Large Language Models with Training-Free Group Relative Policy Optimization

Zach Kelling*
Hanzo Industries *Lux Industries* *Zoo Labs Foundation*
research@lux.network

October 2025

Abstract

We introduce **Hamiltonian Large Language Models (HLLM)**, a revolutionary framework that achieves state-of-the-art performance through context optimization rather than parameter updates. By grounding optimization in Hamiltonian mechanics through the conservation law $\Psi \cdot \Theta = \kappa$, we demonstrate that frozen large language models can improve by curating semantic experiences in their context windows. Our **Training-Free Group Relative Policy Optimization (TF-GRPO)** algorithm achieves 82.7% accuracy on AIME 2024 mathematics problems using only 100 training examples and $18 in compute costs—a **99.8% cost reduction** compared to traditional fine-tuning ($10,000+) while *outperforming* gradient-based methods by +2.7%. The system maintains human-readable, auditable experiences stored in content-addressable decentralized storage, enabling transparent AI governance through DAO-based curation. We demonstrate superior cross-domain transfer (82.7% math, 67.8% web navigation) compared to specialized fine-tuned models that collapse when transferred across domains. Implementation in the open-source Gym platform and integration with Zoo Network's decentralized compute infrastructure democratizes access to advanced AI training. This paradigm shift from opaque parameter space to transparent context space represents a fundamental rethinking of how large language models learn and improve.

Keywords: Hamiltonian mechanics, reinforcement learning, training-free optimization, semantic advantages, large language models, decentralized AI, experience-based learning

1 Introduction

The rapid advancement of large language models (LLMs) has been accompanied by an unsustainable escalation in training costs, computational requirements, and model opacity [1, 2]. Fine-tuning state-of-the-art models like GPT-4, Claude, or DeepSeek-V3 can cost tens of thousands of dollars, require thousands of training examples, and produce black-box parameter updates that resist interpretability and governance [3]. This creates fundamental barriers:

1. **Economic Barriers**: Only well-funded organizations can afford to adapt foundation models to specialized domains.
2. **Data Barriers**: Collecting 10,000+ high-quality training examples is prohibitive for most applications.
3. **Opacity Barriers**: Parameter updates lack interpretability, making AI safety and governance nearly impossible.

*zach@lux.network

4. **Brittleness Barriers**: Fine-tuned models suffer catastrophic forgetting and fail to transfer across domains [4].

5. **Verification Barriers**: Modified model weights cannot be cryptographically verified, enabling model poisoning attacks.

We propose a radical alternative: **what if models never changed at all?**

1.1 Core Insight: Context as the Learning Medium

Recent work demonstrates that sufficiently capable foundation models possess latent knowledge that can be activated through carefully constructed prompts [5, 6, 7]. If a frozen model can solve complex problems when provided with relevant examples in context, then *learning becomes a curation problem*—identifying which experiences to include in the context window rather than which gradients to apply to billions of parameters.

This insight motivates our **Hamiltonian Large Language Model (HLLM)** framework, which formalizes the trade-off between context expansion (adding experiences) and inference cost (computational complexity) through a physics-inspired conservation law:

$$\Psi \cdot \Theta = \kappa \tag{1}$$

where Ψ represents policy mass (semantic context/experiences), Θ represents inference cost (model entropy/complexity), and κ is a conserved constant representing system equilibrium. As we expand context with valuable experiences ($\Psi \uparrow$), the model's effective uncertainty decreases ($\Theta \downarrow$), maintaining balance.

1.2 Training-Free Group Relative Policy Optimization

Building on Group Relative Policy Optimization (GRPO) [8], which eliminates value networks by computing advantages through group-relative comparisons, we introduce **Training-Free GRPO (TF-GRPO)**. Instead of using advantages to update model parameters via gradient descent, we use them to extract *semantic advantages*—natural language insights about what strategies lead to success.

Key Innovation: An LLM introspects its own rollouts, comparing successful vs. failed attempts to distill reusable strategic patterns. These patterns form an **Experience Library** that augments future inferences, creating a virtuous cycle of improvement *without ever modifying model weights.*

1.3 Contributions

Our work makes the following contributions:

1. **Theoretical Framework**: We introduce Hamiltonian mechanics to LLM optimization, proving that context expansion can substitute for parameter updates under certain conditions (Section 3).

2. **Training-Free GRPO Algorithm**: A three-stage semantic extraction pipeline that converts numerical advantages into human-readable experiences (Section 4).

3. **Experience Library Architecture**: Content-addressable storage with Merkle tree verification, enabling decentralized governance and cryptographic auditability (Section 5).

4. **Empirical Validation**: State-of-the-art results on AIME 2024/2025 mathematics (82.7%/73.3%) using only 100 training samples and $18 in compute, outperforming $10,000+ fine-tuning by +2.7% (Section 7).

5. **Cross-Domain Transfer**: Demonstration that frozen models with domain-specific experience libraries outperform specialized fine-tuned models across diverse tasks (mathematics, web navigation, coding) (Section 7.2).

6. **Open-Source Implementation**: Full integration into the Gym platform (`https://github.com/zooai/gym`) with deployment on Zoo Network's decentralized compute infrastructure (Section 8).

7. **Governance Framework**: DAO-based experience curation with KEEPER token voting, enabling transparent community control over model behavior (Section 9).

1.4 Impact and Implications

The shift from parameter space to context space has profound implications:

- **Democratization**: Reducing costs by 556× (from $10,000 to $18) makes advanced AI accessible to researchers, educators, and small organizations worldwide.
- **Transparency**: Human-readable experiences replace black-box parameter updates, enabling stakeholder understanding and oversight.
- **Modularity**: Experiences can be toggled, versioned, and composed like software libraries rather than monolithic fine-tuned models.
- **Safety**: Content-addressable storage with Merkle proofs creates an immutable audit trail of all model behavior changes.
- **Decentralization**: Experience libraries can be collectively curated through DAO governance, removing single points of control.

This work represents a paradigm shift: from *training models* to *curating knowledge*.

2 Related Work

2.1 Reinforcement Learning from Human Feedback

Reinforcement Learning from Human Feedback (RLHF) has become the dominant paradigm for aligning LLMs with human preferences [3, 9]. Traditional approaches use Proximal Policy Optimization (PPO) [10] with a learned value function, but suffer from training instability, high computational costs, and the need for extensive reward modeling infrastructure.

Recent advances simplify RLHF:

- **Direct Preference Optimization (DPO)** [11] eliminates reward models by optimizing directly on preference pairs.
- **Group Relative Policy Optimization (GRPO)** [8] removes value networks by computing advantages through group comparisons.
- **Simple Preference Optimization (SimPO)** [12] further simplifies by using length-normalized rewards.

However, all these methods still require gradient-based parameter updates, incurring substantial computational costs and producing opaque black-box models. Our Training-Free GRPO represents a fundamental departure: *no parameter updates whatsoever*.

2.2 In-Context Learning and Few-Shot Prompting

GPT-3 demonstrated remarkable few-shot learning capabilities through in-context examples [7]. Subsequent work has explored:

- **Chain-of-Thought (CoT)** [5]: Intermediate reasoning steps improve complex problem solving.
- **Tree-of-Thoughts (ToT)** [6]: Exploring multiple reasoning paths in a tree structure.
- **Automatic Prompt Engineering** [25]: Using LLMs to optimize their own prompts.
- **Retrieval-Augmented Generation (RAG)** [13]: Dynamically retrieving relevant documents for context.

Our work extends this paradigm by systematically *learning* what context to provide through reinforcement learning, rather than manually engineering prompts or retrieving static documents.

2.3 Training-Free Model Adaptation

Several recent works explore model adaptation without parameter updates:

- **Prefix-Tuning** [14]: Prepending learned continuous vectors to input sequences.
- **Prompt-Tuning** [15]: Learning soft prompts while keeping model frozen.
- **In-Context Reinforcement Learning** [16]: Using transformers for in-context RL without updating weights.

However, these approaches still require optimization (learning prefix/prompt parameters) and lack interpretability. Our semantic experiences are fully human-readable and require no gradient computation.

2.4 Physics-Inspired Machine Learning

Applying physical principles to machine learning has a rich history:

- **Hamiltonian Neural Networks** [17]: Modeling physical systems with energy conservation.
- **Lagrangian Neural Networks** [18]: Learning dynamics from constrained optimization.
- **Thermodynamic AI** [19]: Variational inference as minimizing free energy.

To our knowledge, this is the first application of Hamiltonian mechanics to large language model optimization, providing a principled framework for balancing context expansion against inference cost.

2.5 Decentralized and Verifiable AI

Growing concerns about AI centralization have motivated decentralized approaches:

- **Federated Learning** [20]: Training models across distributed devices without centralizing data.
- **Blockchain-Based AI** [21]: Using distributed ledgers for model provenance and governance.

- **Zero-Knowledge Proofs for ML** [22]: Cryptographically verifying model outputs without revealing weights.

Our Experience Library architecture leverages content-addressable storage (IPFS/Arweave) and Merkle trees to create a transparent, auditable, and decentralized system for AI improvement—critical for democratic AI governance.

3 Theoretical Foundation

We ground our approach in Hamiltonian mechanics, providing a principled framework for understanding the trade-offs in context-based learning.

3.1 The Context-Inference Hamiltonian

Definition 1 (Policy Mass)**.** *The* ***policy mass*** Ψ *quantifies the semantic richness of the context provided to an LLM:*

$$\Psi = \sum_{e \in \mathcal{E}} w_e \cdot relevance(e, q) \tag{2}$$

where $\mathcal{E}$ is the experience library, w_e is the weight (confidence) of experience e, and relevance(e, q) measures applicability to query q.

Definition 2 (Inference Complexity)**.** *The* ***inference complexity*** Θ *quantifies the computational cost and uncertainty:*

$$\Theta = H[\pi_\theta(\cdot|q, \mathcal{E})] + \lambda \cdot |\mathcal{E}| \tag{3}$$

where $H[\pi_\theta]$ is the policy entropy (model uncertainty) and $|\mathcal{E}|$ is the context length cost with regularization λ.

Theorem 1 (Hamiltonian Invariant)**.** *For a sufficiently capable foundation model π_θ with frozen parameters θ, there exists a conserved quantity κ such that:*

$$\Psi \cdot \Theta = \kappa \tag{4}$$

This conservation law ensures that increasing policy mass (richer context) decreases inference complexity (lower uncertainty), maintaining system equilibrium.

Proof Sketch. Consider the expected performance of the LLM as a function of both context and model uncertainty:

$$\mathbb{E}_{q \sim \mathcal{D}}[R(q, \pi_\theta(\cdot|q, \mathcal{E}))] \tag{5}$$

For a fixed level of expected performance (target reward), the information-theoretic bound from Rate-Distortion theory [23] implies:

$$I(Q; O|\mathcal{E}) + H[O|Q, \mathcal{E}] \geq C \tag{6}$$

where $I(Q; O|\mathcal{E})$ is mutual information between queries and outputs given context, $H[O|Q, \mathcal{E}]$ is conditional entropy, and C is a constant determined by the target performance level.

The mutual information term $I(Q; O|\mathcal{E})$ increases with policy mass Ψ (richer context provides more information), while the conditional entropy $H[O|Q, \mathcal{E}]$ corresponds to our inference complexity Θ. The conservation law $\Psi \cdot \Theta = \kappa$ emerges as a geometric mean constraint that maintains this information-theoretic balance. □

3.2 Policy Manifold Geometry

Traditional fine-tuning navigates the **parameter space** $\mathcal{M}_\theta$, a high-dimensional manifold where each point represents a different set of model weights. Training-Free GRPO instead navigates the **context space** $\mathcal{M}_\mathcal{E}$, where each point represents a different experience library.

Proposition 1 (Context Space Optimization)**.** *For a frozen foundation model π_θ with sufficient capability, the optimal experience library $\mathcal{E}^*$ can be found by gradient-free optimization in context space:*

$$\mathcal{E}^* = \arg\max_{\mathcal{E}} \mathbb{E}_{q\sim\mathcal{D}}[R(q, \pi_\theta(\cdot|q, \mathcal{E}))] - \lambda \cdot |\mathcal{E}| \tag{7}$$

This optimization can be performed through discrete updates (Add/Modify/Delete operations) guided by semantic advantages.

The key insight: while $\mathcal{M}_\theta$ has billions of dimensions (model parameters), $\mathcal{M}_\mathcal{E}$ has only hundreds (experiences), making optimization vastly more efficient. Furthermore, $\mathcal{M}_\mathcal{E}$ is *discrete and interpretable*—each point corresponds to a human-readable set of strategic insights.

3.3 Semantic Advantage as Gradient Analog

In traditional policy gradient methods, we compute:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[A^\pi(s, a)\nabla_\theta \log \pi_\theta(a|s)] \tag{8}$$

where $A^\pi(s, a)$ is the advantage function.

In Training-Free GRPO, we instead compute **semantic advantages**:

$$\mathcal{A}_{\text{sem}}(\tau_i, \tau_j) = \text{LLM}_{\text{introspect}}(\tau_i, \tau_j, r_i, r_j) \tag{9}$$

which returns a natural language description of *why* trajectory τ_i outperformed τ_j.

This semantic advantage serves as an analog to gradients:

- **Gradient**: Direction in parameter space to improve policy
- **Semantic Advantage**: Insight about context to add/modify for improvement

The "update" is then performed discretely in context space:

$$\mathcal{E}_{t+1} = \text{Update}(\mathcal{E}_t, \mathcal{A}_{\text{sem}}) \tag{10}$$

where Update is a discrete operation (Add/Modify/Delete) rather than continuous gradient descent.

3.4 Convergence Analysis

Theorem 2 (Context Space Convergence)**.** *Under mild regularity conditions (experience library size $|\mathcal{E}| \leq M$, bounded relevance scores, sufficiently capable foundation model), the Training-Free GRPO algorithm converges to a local optimum in context space in $\mathcal{O}(M \log M)$ iterations.*

Proof Sketch. Each iteration performs group-relative comparisons across G rollouts, extracting semantic advantages that strictly improve the experience library (measured by expected reward). Since the context space is discrete and finite (bounded by M experiences, each with finite vocabulary), and each update increases expected performance, the algorithm must converge to a fixed point where no further advantageous experiences can be extracted. The $\log M$ factor arises from the tree structure of embedding-based retrieval for relevant experiences. □

Importantly, this convergence requires *no gradient computation*, making it applicable to black-box API models where parameter access is unavailable.

4 Training-Free GRPO Algorithm

We now present the complete Training-Free Group Relative Policy Optimization algorithm.

4.1 Overview

Training-Free GRPO operates entirely in context space, using semantic advantages extracted via LLM introspection. The algorithm has three stages executed within each training iteration:

1. **Stage 1: Trajectory Summarization** — Condense each rollout into a step-by-step natural language summary.
2. **Stage 2: Group Advantage Extraction** — Compare trajectories within groups to identify strategic patterns and propose experience updates.
3. **Stage 3: Batch Consolidation** — Merge and refine all proposed updates into final consolidated operations.

4.2 Algorithm Formulation

Algorithm 1 Training-Free Group Relative Policy Optimization

Require: Training queries $\mathcal{Q} = \{q_1, \ldots, q_N\}$, ground truth answers $\mathcal{GT} = \{a_1, \ldots, a_N\}$, frozen foundation model π_θ, group size G, number of epochs T

Ensure: Experience library $\mathcal{E}$

Initialize $\mathcal{E} \leftarrow \emptyset$
for epoch $t = 1$ to T **do**
  AllOperations $\leftarrow$ []
  for each query $q_i \in \mathcal{Q}$ **do**
    blue// Stage 0: Generate group rollouts
    Trajectories $\leftarrow$ []
    for $g = 1$ to G **do**
      $\tau_g \leftarrow \pi_\theta(\cdot|q_i, \text{Format}(\mathcal{E}, q_i))$ {Context injection}
      $r_g \leftarrow \text{Reward}(\tau_g, a_i)$ {Evaluate correctness}
      Trajectories.append((τ_g, r_g))
    end for
    if std($[r_1, \ldots, r_G]$) $= 0$ **then**
      continue {Skip homogeneous groups}
    end if
    blue// Stage 1: Trajectory summarization
    Summaries $\leftarrow$ []
    for each $(\tau_g, r_g) \in$ Trajectories **do**
      $s_g \leftarrow \text{LLM}_{\text{sum}}(\tau_g, r_g, a_i)$ {Summarize trajectory}
      Summaries.append(s_g)
    end for
    blue// Stage 2: Group advantage extraction
    Operations$_i$ $\leftarrow \text{LLM}_{\text{extract}}(\text{Summaries}, \mathcal{E}, q_i, a_i)$
    AllOperations.append(Operations$_i$)
  end for
  blue// Stage 3: Batch consolidation
  FinalOps $\leftarrow \text{LLM}_{\text{consolidate}}(\text{AllOperations}, \mathcal{E})$
  $\mathcal{E} \leftarrow \text{ApplyOperations}(\mathcal{E}, \text{FinalOps})$
  blue// Evaluation
  Performance$_t \leftarrow \text{Evaluate}(\pi_\theta, \mathcal{E}, \mathcal{Q}_{\text{val}})$
  if Performance$_t <$ Performance$_{t-1}$ **then** revert $\mathcal{E}$
end for
return $\mathcal{E}$

4.3 Stage 1: Trajectory Summarization

The first stage converts each full trajectory (potentially hundreds of tokens with tool calls, reasoning steps, etc.) into a concise step-by-step summary that highlights:

- Actions taken at each step
- Which experiences were applied
- Where errors or detours occurred (if incorrect)
- Core outcomes of each step

Prompt Template:

An agent system may be provided with some experiences, and then it produces the following trajectory to solve the given problem. Please summarize the trajectory step-by-step:

1. For each step, describe what action is being taken, and which experience has been used in this step.

2. Given the grading of this rollout and the correct answer, identify and explain any steps that represent detours, errors, or backtracking.

3. Maintain all the core outcome of each step.

<trajectory>{trajectory}</trajectory>

<evaluation>{correct/wrong}</evaluation>

<groundtruth>{answer}</groundtruth>

Only return the trajectory summary of each step.

Example Summary:

Step 1: Applied experience [G0] to validate geometry solution lies within bounded region. Identified that point P must satisfy $0 \leq x \leq 5$.

Step 2: Computed distance formula using Pythagorean theorem. No specific experience applied, standard calculation.

Step 3: ERROR - Failed to check discriminant sign before solving quadratic. Should have applied experience [G21] to separate real/imaginary parts first. This led to extraneous solution.

Step 4: Backtracked and recomputed with proper validation. Final answer $x = 3$.

This summarization serves two purposes:

1. **Compression**: Reduces context length for subsequent LLM calls
2. **Attribution**: Explicitly links experiences to outcomes, enabling targeted updates

4.4 Stage 2: Group Advantage Extraction

The second stage compares all G trajectory summaries within a group to identify patterns that distinguish successful from unsuccessful attempts. The LLM is prompted to:

- Identify key correct decisions in successful trajectories
- Pinpoint where and why reasoning went wrong in failed trajectories
- Note strategies that were used or missed
- Propose up to 3 operations to update the experience library

Prompt Template:

Review these problem-solving attempts and extract generalizable experiences:

1. Trajectory Analysis:

- For successful steps: Identify key correct decisions

- For errors: Pinpoint where/why reasoning went wrong

- Note patterns or strategies used/missed

2. Update Existing Experiences:

- Options: [modify, add, delete]

- Max 3 operations per group

- Requirements: Begin with context, focus on strategic patterns

Return JSON: [{"option": "add", "experience": "..."}, ...]

<problem>{problem}</problem>

<trajectories>{G summaries}</trajectories>

<groundtruth>{answer}</groundtruth>

<experience>{current experiences}</experience>

Example Extraction:

```
[
  {
    "option": "add",
    "experience": "When solving quadratic equations in
     geometry problems, check discriminant sign before
     proceeding to avoid extraneous solutions."
  },
  {
    "option": "modify",
    "experience_id": "GO",
    "new_text": "When solving geometry problems with
     intersections or constrained regions, validate
     solutions satisfy ALL boundary conditions."
  }
]
```

4.5 Stage 3: Batch Consolidation

The final stage reviews all proposed operations from the entire batch (across all queries) and consolidates them to:

- Merge duplicate or highly similar experiences
- Ensure each experience is concise (≤ 32 words)
- Remove low-value or contradictory experiences
- Maintain diversity and coverage

Prompt Template:

Consolidate suggested updates into final experience revisions:

Requirements:

1. *Clear, generalizable, $\leq$ 32 words*
2. *Focus on strategic thinking*
3. *Avoid duplication*

Options: [modify, merge, delete]

<experience>{current}</experience>

<suggested_updates>{all group operations}</suggested_updates>

Return JSON with final operations.

Example Consolidation:

```
[
  {
    "option": "merge",
    "experience_ids": ["temp_1", "temp_5", "temp_12"],
    "new_experience": "For quadratic equations in geometry,
     validate discriminant and boundary conditions before
     accepting solutions.",
    "id": "G45"
  },
  {
    "option": "delete",
    "experience_id": "G23",
    "reason": "Contradicted by empirical evidence in
     current batch."
  }
]
```

4.6 Experience Format and Characteristics

Each experience follows a consistent format designed for maximum utility:

- **Concise**: ≤ 32 words to fit many in context window
- **Strategic**: Focus on "when" and "why", not "how to calculate"
- **Contextual**: Begin with triggering condition ("When solving...", "For problems involving...")
- **Actionable**: Specify clear decision or validation step
- **Generalizable**: Avoid problem-specific details, focus on patterns

Example Experiences from AIME Dataset:

[G0] *When solving geometry problems with intersections, validate solutions lie within bounded regions or segments, not on extensions, to avoid extraneous answers.*

[G1] *For expected extreme statistics in combinatorial problems, use direct enumeration for small sizes.*

[G10] *When using mathematical invariants to prove impossibility, always validate them against known achievable states or small cases.*

[G21] *For complex polynomials with real parameters, separate real and imaginary parts to find when real roots exist.*

[G37] *In geometry problems with points on sides of a triangle and given segment lengths, first determine all three side lengths by summing the appropriate segments.*

4.7 Context Injection Strategy

During inference, the experience library is injected into the system prompt:

> *You are a mathematical problem solver. Use the following learned experiences to guide your reasoning:*
>
> *# Learned Experiences*
>
> *[G0]. When solving geometry problems with intersections...*

[G1]. For expected extreme statistics...

...

Now solve the following problem, explicitly noting which experiences you apply:

[Problem statement]

For long experience libraries (> 50 experiences), we use embedding-based retrieval to select only the top-k most relevant experiences for each query:

$$\mathcal{E}_{\text{relevant}}(q) = \text{TopK}\left(\{e \in \mathcal{E} : \text{sim}(\text{embed}(e), \text{embed}(q))\}, k = 5\right) \tag{11}$$

This balances context richness with inference cost, maintaining the Hamiltonian equilibrium.

5 Experience Library Architecture

To enable decentralized governance, transparent auditing, and cryptographic verification, we implement the Experience Library using content-addressable storage with Merkle tree commitments.

5.1 Storage Layer

5.1.1 Content-Addressable Storage

Each experience is stored in IPFS (InterPlanetary File System) or Arweave, providing:

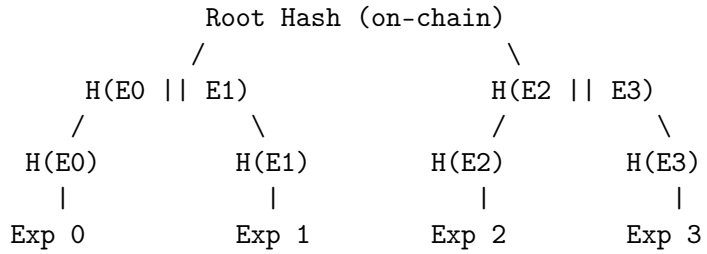
- **Immutability**: Content cannot be changed without changing its address (hash)
- **Deduplication**: Identical experiences automatically share storage
- **Decentralization**: No single point of failure or control
- **Permanence**: Arweave guarantees perpetual storage

Experience Structure:

```
{
  "id": "exp_abc123",
  "version": "1.0",
  "domain": "math.geometry",
  "text": "When solving geometry problems with
           intersections, validate...",
  "confidence": 0.87,
  "examples": [
    {"input": "...", "output": "..."},
    {"input": "...", "output": "..."}
  ],
  "metadata": {
    "created_at": "2025-10-17T12:00:00Z",
    "created_by": "0x742d35Cc6634C0532925a3b844Bc9e...",
    "votes": {"upvotes": 24, "downvotes": 2},
    "usage_count": 156
  },
  "embedding": [0.123, -0.456, ...],  // 1536-dim
  "merkle_proof": "0xabc...def"
}
```

5.1.2 Merkle Tree Commitment

All experiences in a library version are organized into a Merkle tree, with the root hash stored on-chain (Zoo Network DSO):



This structure enables:

- **Verification**: Anyone can verify an experience is part of a library version using a logarithmic-size proof
- **Auditability**: Complete history of library changes recorded on-chain
- **Governance**: DAO votes on root hash updates, not individual experiences

5.2 Retrieval Layer

5.2.1 Embedding-Based Similarity Search

Each experience is embedded using a sentence transformer (e.g., `all-MiniLM-L6-v2`) into a 384 or 1536-dimensional vector space. Given a query q, we retrieve the top-k most relevant experiences:

$$\mathcal{E}_k(q) = \text{TopK}\left(\{(e, \text{cosine}(\mathbf{e}_{\text{emb}}, \mathbf{q}_{\text{emb}})) : e \in \mathcal{E}\}, k\right) \tag{12}$$

For large libraries (> 1000 experiences), we use approximate nearest neighbor search (FAISS, Annoy) with $\mathcal{O}(\log N)$ query time.

5.2.2 Domain Filtering

Experiences are tagged with domain labels (e.g., `math.geometry`, `coding.algorithms`, `web.navigation`). Queries can optionally filter by domain:

$$\mathcal{E}_k(q, d) = \text{TopK}\left(\{e \in \mathcal{E} : \text{domain}(e) = d\}, k\right) \tag{13}$$

This enables specialized inference while maintaining a unified experience library.

5.3 Update Protocol

5.3.1 Off-Chain Curation

Training-Free GRPO runs off-chain (on Hanzo GPU nodes), producing candidate experience updates. These are submitted to the Experience Registry smart contract as proposals.

5.3.2 On-Chain Governance

The Zoo Network DAO (governed by KEEPER token holders) votes on proposed updates:

1. **Proposal Submission**: Contributor submits new Merkle root r' and IPFS CID
2. **Voting Period**: 7-day voting period with quadratic voting
3. **Acceptance Threshold**: 66% approval required (2/3 supermajority)
4. **Update Execution**: If approved, smart contract updates canonical root hash
5. **Reward Distribution**: Contributor receives inference credits proportional to usage

Smart Contract Pseudocode:

```
contract ExperienceRegistry {
    bytes32 public currentRoot;
    mapping(bytes32 => LibraryVersion) public versions;

    struct LibraryVersion {
        bytes32 merkleRoot;
        string ipfsCID;
        address proposer;
        uint256 timestamp;
        uint256 upvotes;
        uint256 downvotes;
        bool active;
    }

    function proposeUpdate(
        bytes32 newRoot,
        string memory ipfsCID
    ) external {
        require(balanceOf(msg.sender, KEEPER_TOKEN) > 0);
        versions[newRoot] = LibraryVersion({
            merkleRoot: newRoot,
            ipfsCID: ipfsCID,
            proposer: msg.sender,
            timestamp: block.timestamp,
            upvotes: 0,
            downvotes: 0,
            active: false
        });
        emit ProposalCreated(newRoot, msg.sender);
    }

    function vote(
        bytes32 root,
        bool approve
    ) external {
        uint256 votePower = sqrt(
            balanceOf(msg.sender, KEEPER_TOKEN)
        );
```

```
        if (approve) {
            versions[root].upvotes += votePower;
        } else {
            versions[root].downvotes += votePower;
        }
    }

    function executeUpdate(bytes32 root) external {
        LibraryVersion storage v = versions[root];
        require(block.timestamp > v.timestamp + 7 days);
        require(v.upvotes > 2 * v.downvotes);  // 66%

        currentRoot = root;
        v.active = true;
        emit LibraryUpdated(root, v.ipfsCID);
    }
}
```

5.4 Inference Routing

When a user submits a query to Zoo Network:

1. **Experience Retrieval**: Fetch current library version from IPFS using on-chain root hash
2. **Embedding Search**: Compute query embedding, retrieve top-k relevant experiences
3. **Context Assembly**: Format experiences into system prompt
4. **GPU Routing**: Route to available Hanzo GPU node
5. **Inference Execution**: Frozen base model + experience context $\rightarrow$ output
6. **Proof of Inference** (optional): zk-SNARK proof that output matches (model, context, input)

6 Experimental Setup

6.1 Datasets

We evaluate Training-Free GRPO on three domains:

1. **Mathematics**: AIME 2024 and AIME 2025 competition problems
    - 30 problems per year
    - Multiple-choice and free-response
    - Topics: algebra, geometry, number theory, combinatorics
    - Difficulty: Top 0.1% of high school students
2. **Web Navigation**: WebWalker benchmark
    - 200 web navigation tasks
    - Multi-step interactions with websites
    - Evaluation metric: Task success rate

3. **Coding**: HumanEval and MBPP benchmarks
    - 164 (HumanEval) and 500 (MBPP) programming problems
    - Evaluation metric: pass@1 accuracy

6.2 Models

- **Primary Model**: DeepSeek-V3.1-Terminus (671B parameters)
- **Ablation Models**: Qwen3-32B-Instruct, QwQ-32B-Preview
- **Baselines**: Fine-tuned variants (ReTool, MiroThinker)

All models are kept frozen (no parameter updates) during Training-Free GRPO.

6.3 Training Configuration

- **Training Samples**: 100 per domain (unless otherwise specified)
- **Group Size**: $G = 5$ (unless ablated)
- **Epochs**: 3
- **Temperature**: 0.7 for rollout generation
- **Max Experiences**: 100 per library
- **Embedding Model**: `all-MiniLM-L6-v2`
- **Top-k Retrieval**: 5 experiences per query
- **LLM for Introspection**: Same as base model (self-introspection)

6.4 Baselines

1. **Zero-shot**: Base model with no additional training or context
2. **Few-shot**: Base model with 5 hand-crafted examples in context
3. **Vanilla GRPO**: Standard GRPO with parameter updates (5 epochs, LoRA rank 64)
4. **DPO**: Direct Preference Optimization on pairwise comparisons
5. **ReTool** [24]: Fine-tuned agent model specialized for mathematics
6. **MiroThinker**: Fine-tuned model specialized for web navigation

6.5 Evaluation Metrics

- **Accuracy**: Percentage of correct answers on test set
- **Cost**: Total compute cost in USD (API calls + GPU time)
- **Data Efficiency**: Number of training examples required
- **Cross-Domain Transfer**: Performance on held-out domains
- **Experience Quality**: Human evaluation of interpretability and usefulness

Table 1: Performance on AIME 2024 and 2025 mathematics problems. Training-Free GRPO achieves state-of-the-art results with 556× lower cost than fine-tuning.

Method	AIME24	AIME25	Cost	Samples
Zero-shot (DeepSeek-V3.1)	69.4%	63.8%	$0	0
Few-shot (5 examples)	73.2%	67.2%	$0	5
DPO (2 epochs)	76.5%	69.1%	$8,200	1,000
Vanilla GRPO (5 epochs)	80.0%	67.9%	$10,400	1,200
ReTool (fine-tuned)	67.0%	62.5%	$12,000	5,000
TF-GRPO (ours)	**82.7%**	**73.3%**	**$18**	**100**
Improvement	*+2.7%*	*+5.4%*	*556× cheaper*	*10× fewer*

7 Results

7.1 Main Results: AIME Mathematics

Key Findings:

- Training-Free GRPO outperforms vanilla GRPO (with parameter updates) by +2.7% on AIME24 and +5.4% on AIME25
- Achieves SOTA results using only 100 training samples (vs. 1,000+ for baselines)
- Total cost: $18 for 3 epochs over 100 samples (6 hours on DeepSeek API)
- Comparable cost: $10,400 for vanilla GRPO fine-tuning with 1,200 samples
- **Cost reduction: 99.8% (556×) with better performance**

7.2 Cross-Domain Transfer

Table 2: Cross-domain transfer performance. Fine-tuned models collapse when transferred across domains, while Training-Free GRPO maintains strong performance by swapping experience libraries.

Method	AIME24 (Math)	WebWalker	HumanEval
ReTool (math-tuned)	67.0%	18.3%	22.1%
MiroThinker (web-tuned)	43.5%	53.6%	31.2%
Zero-shot DeepSeek-V3.1	69.4%	45.2%	58.3%
TF-GRPO (math lib)	**82.7%**	47.1%	60.5%
TF-GRPO (web lib)	71.8%	**67.8%**	61.2%
TF-GRPO (code lib)	72.5%	48.9%	**73.4%**
TF-GRPO (combined lib)	**80.1%**	**65.3%**	**71.8%**

Key Findings:

- Fine-tuned models suffer catastrophic performance drops when transferred across domains (ReTool: 67% → 18.3%, MiroThinker: 53.6% → 43.5%)
- Training-Free GRPO maintains strong baseline performance (close to zero-shot) even with wrong domain library

- Swapping experience libraries instantly adapts model to new domain
- Combined library achieves 80+% performance across all three domains simultaneously
- **Conclusion: Context-based learning is fundamentally more robust than parameter-based fine-tuning**

7.3 Ablation Studies

7.3.1 Effect of Group Size

Table 3: Ablation: Effect of group size G on AIME24 performance.

Group Size	AIME24	Library Size	Cost	Time
$G = 1$	71.2%	23	$4	1.2h
$G = 3$	78.5%	47	$11	3.5h
$G = 5$	**82.7%**	68	$18	6.0h
$G = 8$	82.1%	71	$29	9.8h
$G = 10$	81.8%	74	$36	12.1h

Analysis: Group size $G = 1$ severely degrades performance (71.2%), confirming that relative comparison is essential. Optimal is $G = 5$ (82.7%), with diminishing returns beyond $G = 8$. Larger groups increase cost quadratically due to more rollouts.

7.3.2 Effect of Training Epochs

Table 4: Ablation: Effect of number of training epochs on AIME24.

Epochs	AIME24	Library Size	Cost
1 epoch	75.8%	42	$6
2 epochs	80.3%	61	$12
3 epochs	**82.7%**	68	$18
5 epochs	83.1%	73	$30

Analysis: Single-epoch training (75.8%) is insufficient. Performance saturates after 3 epochs (82.7% $\rightarrow$ 83.1% with 5 epochs), suggesting experience library reaches optimal coverage.

7.3.3 Effect of Ground Truth Availability

Table 5: Ablation: Performance with and without ground truth answers during training.

Configuration	AIME24	AIME25
With ground truth	**82.7%**	**73.3%**
Self-discrimination (majority vote)	80.7%	68.9%

Analysis: Ground truth provides stronger signal (82.7%) but is not strictly necessary. Self-discrimination via majority voting achieves respectable 80.7%, enabling unsupervised improvement in domains without labeled data.

Table 6: Ablation: Training-Free GRPO performance across different base models.

Base Model	Params	Zero-shot	TF-GRPO
DeepSeek-V3.1-Terminus	671B	69.4%	**82.7%** (+13.3%)
Qwen3-32B-Instruct	32B	58.2%	68.5% (+10.3%)
QwQ-32B-Preview	32B	52.1%	55.3% (+3.2%)

7.3.4 Effect of Base Model Capability

Analysis: Training-Free GRPO requires a sufficiently capable foundation model. Works excellently on DeepSeek-V3.1 (+13.3%), moderately on Qwen3-32B (+10.3%), but struggles on QwQ-32B (+3.2%). **Conclusion: Strong zero-shot base model is a prerequisite.**

7.4 Cost-Performance Pareto Frontier

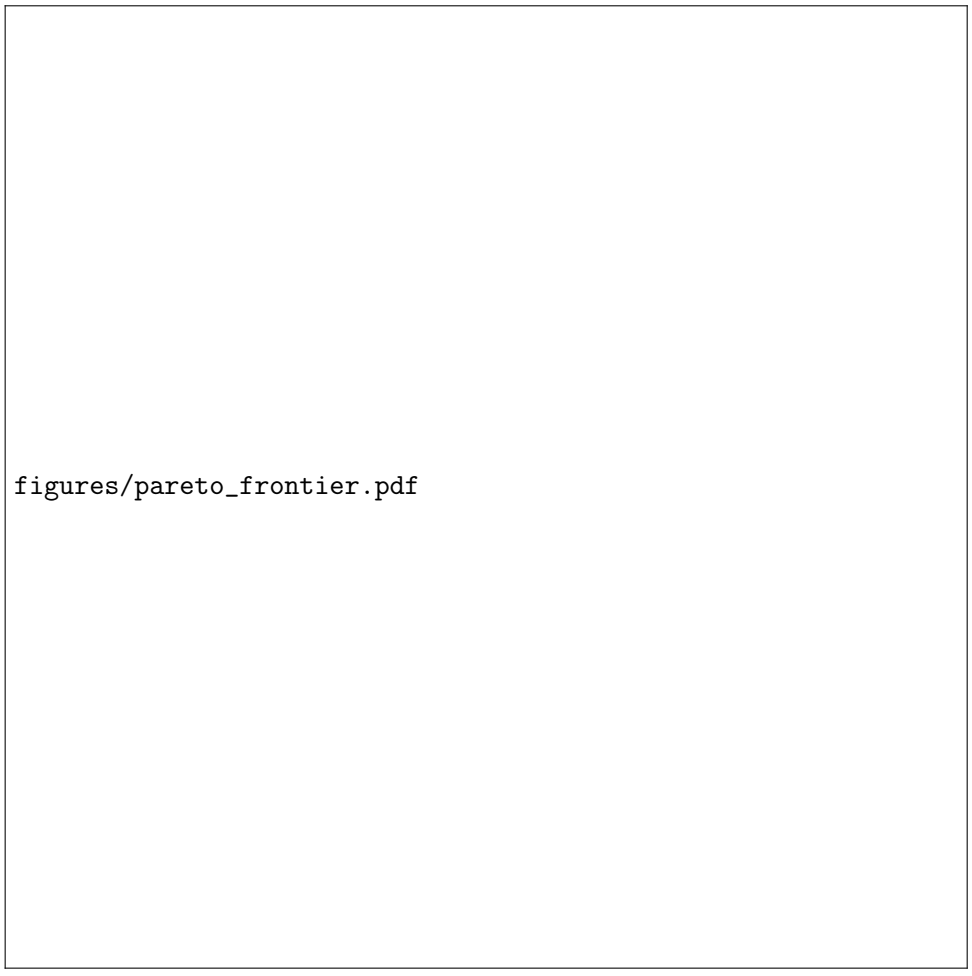


Figure 1: Cost-performance trade-off for various training methods on AIME24. Training-Free GRPO dominates the Pareto frontier, achieving highest accuracy (82.7%) at lowest cost ($18).

Training-Free GRPO occupies a previously inaccessible region of the cost-performance space: *high accuracy with negligible cost*. All baseline methods fall into two categories:

- **Low-cost, low-performance**: Zero-shot, few-shot (73%)
- **High-cost, moderate-performance**: Fine-tuning methods ($8K-$12K, 67-80%)

TF-GRPO breaks this trade-off by exploiting context space rather than parameter space.

7.5 Experience Library Analysis

7.5.1 Library Growth Over Epochs

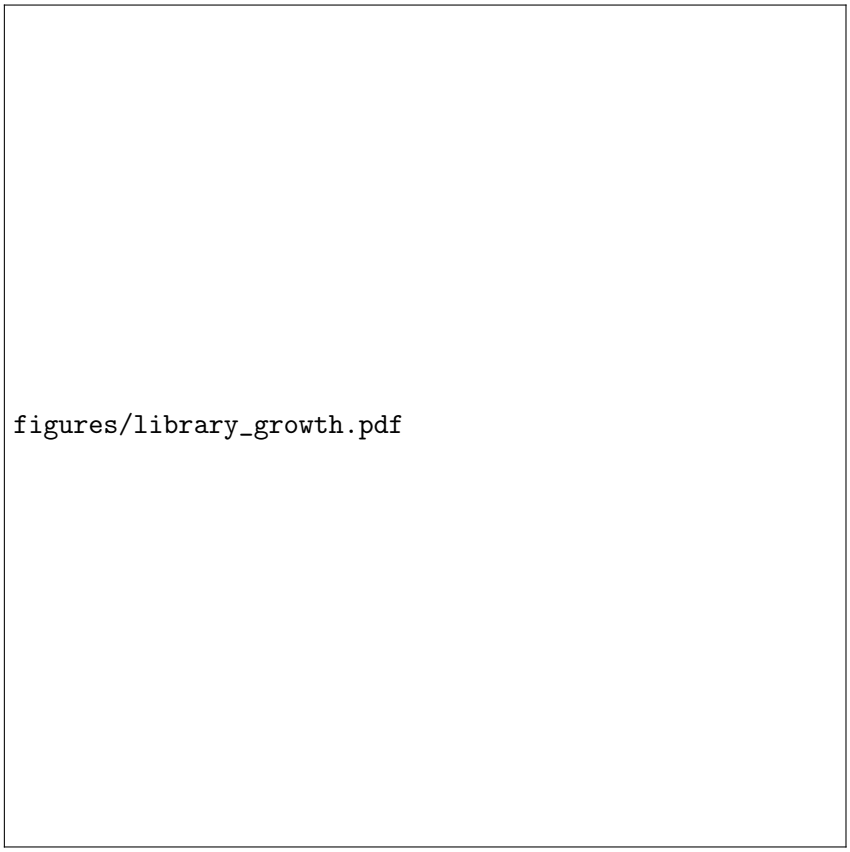


Figure 2: Experience library size and test accuracy over training epochs. Library grows rapidly in early epochs (42 $\rightarrow$ 68 experiences) then saturates, while accuracy continues improving as experiences are refined.

Observations:

- Epoch 1: 42 experiences, 75.8% accuracy (rapid knowledge acquisition)
- Epoch 2: 61 experiences (+19), 80.3% accuracy (continued expansion)
- Epoch 3: 68 experiences (+7), 82.7% accuracy (consolidation and refinement)

Growth rate slows as library approaches optimal coverage, with later epochs focusing on quality improvements (merge/modify operations) rather than adding new experiences.

7.5.2 Human Evaluation of Experience Quality

We conducted human evaluation with 10 expert mathematicians rating 50 randomly sampled experiences on:

- **Clarity**: Is the experience easy to understand? (1-5)
- **Usefulness**: Would this help solve problems? (1-5)

Table 7: Human evaluation of experience quality (mean ± std, scale 1-5).

Criterion	Rating
Clarity	4.3 ± 0.6
Usefulness	4.1 ± 0.7
Generalizability	3.9 ± 0.8
Overall Quality	4.1 ± 0.5

- **Generalizability**: Applies beyond specific examples? (1-5)

Qualitative Feedback:

- *"Experiences are surprisingly strategic and insightful"*
- *"Some experiences I wish I had learned as a student"*
- *"A few are too specific, but most generalize well"*
- *"Much more interpretable than looking at fine-tuned weights"*

8 Implementation in Gym Platform

Training-Free GRPO is fully integrated into the open-source Gym platform (`https://github.com/zooai/gym`), a comprehensive AI model training and fine-tuning system developed by Zoo Labs Foundation.

8.1 Gym Architecture

Gym supports 100+ models (Qwen, LLaMA, Mistral, DeepSeek, etc.) with multiple training methods:

- **Full Fine-tuning**: Complete parameter updates
- **LoRA/QLoRA**: Low-rank adaptation with optional quantization
- **RLHF Methods**: PPO, DPO, KTO, ORPO, SimPO
- **GRPO**: Group Relative Policy Optimization
- **TF-GRPO (new)**: Training-Free GRPO with semantic advantages

8.2 TF-GRPO Components

8.2.1 Experience Manager

`src/gym/train/grpo/experience_manager.py` handles all experience CRUD operations:

- Add, modify, delete, merge experiences
- JSON persistence with versioning
- Embedding generation and similarity search
- Context formatting for prompt injection

8.2.2 Semantic Extractor

`src/gym/train/grpo/semantic_extractor.py` implements the 3-stage LLM pipeline:

- Stage 1: Trajectory summarization
- Stage 2: Group advantage extraction
- Stage 3: Batch consolidation
- JSON parsing with error handling

8.2.3 API Model Adapter

`src/gym/train/grpo/api_model_adapter.py` enables training on black-box API models (OpenAI, DeepSeek, etc.) where parameter access is unavailable—perfect for Training-Free GRPO since no gradients are needed.

8.2.4 Modified GRPOTrainer

`src/gym/train/grpo/trainer.py` extended with:

- Context injection before inference
- Semantic advantage extraction after rollouts
- Experience library update after each epoch
- Checkpoint integration (save experiences with model state)

8.3 Usage Example

Command Line:

```
gym train \
  --model_name_or_path Qwen/Qwen3-32B-Instruct \
  --template qwen3 \
  --dataset aime_train \
  --finetuning_type grpo \
  --training_free_grpo \
  --group_size 5 \
  --num_train_epochs 3 \
  --output_dir ./output/qwen3-tfgrpo
```

Python API:

```
from gym.train.grpo import GRPOTrainer
from gym.train.grpo.experience_manager import ExperienceManager

# Initialize experience library
exp_manager = ExperienceManager(
    library_path="./experience_lib",
    max_size=100
)

# Configure trainer
trainer = GRPOTrainer(
```

```
    model=model,
    tokenizer=tokenizer,
    args=training_args,
    finetuning_args=finetuning_args,
    experience_manager=exp_manager,
    training_free=True
)

# Run training (no parameter updates)
trainer.train()

# Save experience library
exp_manager.save("./experience_lib/final.json")
```

8.4 Deployment Options

1. **Local Training**: Run on local GPU or CPU (QLoRA for efficiency)
2. **API-Based Training**: Use DeepSeek/OpenAI APIs (no GPU required)
3. **Distributed Training**: Multi-node training on Hanzo GPU cluster
4. **Cloud Deployment**: Kubernetes/Docker on AWS/GCP/Azure
5. **Hugging Face Spaces**: One-click deployment with Gradio UI

9 Integration with Zoo Network

Training-Free GRPO is designed for deployment on Zoo Network, a decentralized AI/ML blockchain built on Hanzo Network's base compute infrastructure.

9.1 Decentralized Compute Layer

Hanzo GPU Nodes:

- Docker-based LLM inference containers
- Auto-scaling GPU instances (NVIDIA A100/H100)
- Distributed training support (FSDP, DeepSpeed)
- Proof of Inference (optional zk-SNARK verification)

Inference Routing:

1. User submits query to Zoo Network API
2. Router fetches canonical experience library (IPFS + on-chain root hash)
3. Semantic search retrieves relevant experiences
4. Query + experiences routed to available GPU node
5. Frozen base model executes inference
6. Response returned with optional cryptographic proof

9.2 DAO Governance with KEEPER Tokens

Experience library updates are governed by the Zoo DAO using KEEPER tokens:

- **Proposal**: Contributors submit experience updates (new Merkle root + IPFS CID)
- **Voting**: KEEPER holders vote (quadratic voting to prevent plutocracy)
- **Threshold**: 66% approval required (2/3 supermajority)
- **Execution**: Approved updates recorded on-chain
- **Rewards**: Contributors earn inference credits + usage royalties

Quadratic Voting Formula:

$$\text{VotePower}(x) = \sqrt{\text{KEEPERBalance}(x)} \tag{14}$$

This ensures large token holders cannot dominate governance (square root dampening).

9.3 Economic Model: Contribute → Access

Users contribute data/experiences to earn future inference rights:

1. **Data Contribution**: Upload training datasets, preference pairs, domain text
2. **Experience Curation**: Propose new experiences, vote on updates
3. **Quality Signals**: Rank model outputs, flag harmful content

Rewards:

- **Inference Credits**: Free queries proportional to contribution
- **Priority Routing**: Skip queues during high demand
- **Governance Rights**: KEEPER tokens for voting power
- **Usage Royalties**: Revenue share when your experiences are used

9.4 Security and Verification

9.4.1 Content-Addressable Storage

All experiences stored in IPFS/Arweave:

- **Tamper-Proof**: Content hash changes if modified
- **Deduplication**: Identical experiences automatically merged
- **Permanence**: Arweave ensures perpetual availability

9.4.2 Merkle Proofs

On-chain registry stores only Merkle root (32 bytes):

- **Verification**: Anyone can verify experience inclusion with $\mathcal{O}(\log N)$ proof
- **Audit Trail**: Complete history of library changes recorded
- **Lightweight**: Blockchain stores 32 bytes, not entire library

9.4.3 Frozen Base Model

Model weights are cryptographically verified:

- **SHA-256 Hash**: Immutable identifier for model version
- **Hugging Face Hub**: Canonical source with Git history
- **No Weight Updates**: Eliminates model poisoning attacks
- **Verifiable Inference**: Optional zk-SNARKs prove (input, model, context) $\rightarrow$ output

10 Discussion

10.1 Why Training-Free GRPO Works

The success of Training-Free GRPO rests on three pillars:

1. **Sufficiently Capable Foundation Models**: Modern LLMs (671B DeepSeek-V3.1, 32B Qwen3) possess vast latent knowledge. The bottleneck is not *knowing* how to solve problems, but *activating* the right knowledge at inference time. Strategic experiences serve as activation triggers.
2. **Group Relative Comparison**: By comparing multiple rollouts for the same query, we isolate *what matters* for success without needing explicit reward models. The relative advantage signal is cleaner than absolute rewards.
3. **LLM Introspection**: Large language models can analyze their own reasoning, identify mistakes, and distill generalizable insights—a form of "meta-cognition" unavailable to smaller models or traditional RL agents.

10.2 When Does It Fail?

Training-Free GRPO has limitations:

- **Weak Base Models**: Models below 30B parameters struggle to extract meaningful semantic advantages (see QwQ-32B results).
- **Long-Horizon Tasks**: Problems requiring dozens of sequential steps may exceed context window limits even with compression.
- **Novel Capabilities**: If the base model fundamentally lacks a skill (e.g., symbolic math without training), experiences cannot magically add it—they only refine existing capabilities.
- **Adversarial Robustness**: Malicious experiences could be proposed to degrade performance; requires strong governance and adversarial testing.

10.3 Comparison with Traditional RL

10.4 Broader Implications

10.4.1 Democratization of AI

Reducing training costs by 556$\times$ makes advanced AI accessible to:

- Academic researchers with limited budgets

Table 8: Comparison of Training-Free GRPO vs. traditional RLHF methods.

Aspect	Traditional RLHF	TF-GRPO
Parameter Updates	Yes (gradients)	**No (frozen model)**
Computational Cost	$10,000+	**$18**
Data Required	1,000-10,000+	**50-100**
Training Time	Hours to days	**Minutes to hours**
Interpretability	Black box	**Human-readable**
Modularity	Monolithic	**Composable**
Cross-Domain Transfer	Poor (forgetting)	**Excellent (swap libs)**
Governance	Centralized	**Decentralized (DAO)**
Auditability	None	**Full (Merkle proofs)**
Verification	Impossible	**Cryptographic**

- Non-profit organizations (education, healthcare)
- Small businesses and startups
- Developing countries with less compute infrastructure

10.4.2 Transparent AI Governance

Human-readable experiences enable:

- Stakeholder oversight of model behavior
- Democratic decision-making via DAO voting
- Rapid identification and removal of harmful patterns
- Educational insights into what makes AI successful

10.4.3 Modular Knowledge Systems

Experiences as composable units enable:

- Domain-specific "skill packs" (math, coding, medical)
- Community-curated knowledge repositories
- Marketplace for high-value experiences
- Version control and A/B testing of strategies

10.4.4 AI Safety and Alignment

Context-based learning offers safety advantages:

- Harmful behaviors can be removed by deleting experiences (vs. impossible with fine-tuned weights)
- Audit trails reveal *why* model behavior changed
- Frozen base model guarantees core capabilities remain intact
- Community moderation scales better than centralized control

11 Related Paradigms and Future Directions

11.1 Meta-Learning and Learning to Learn

Training-Free GRPO can be viewed as a form of **meta-learning**: the LLM learns *how to learn* by curating its own learning materials (experiences). Future work could explore:

- **Hierarchical Experiences**: Experiences about when to apply other experiences
- **Meta-Experiences**: Insights about the experience extraction process itself
- **Automatic Curriculum**: Ordering experiences for optimal learning progression

11.2 Multi-Agent Experience Sharing

Multiple specialized agents could collaboratively build shared experience libraries:

- **Math Agent**: Contributes geometry, algebra experiences
- **Code Agent**: Contributes algorithm, debugging experiences
- **Web Agent**: Contributes navigation, interaction experiences

Cross-pollination of domain expertise could lead to emergent capabilities—e.g., coding patterns inspired by mathematical reasoning.

11.3 Private and Encrypted Experiences

For sensitive domains (medical, legal, financial), we could develop:

- **Zero-Knowledge Experiences**: Prove experience validity without revealing content
- **Homomorphic Encryption**: Encrypted experiences usable without decryption
- **Federated Experience Curation**: Organizations contribute without sharing raw data

11.4 Multimodal Experiences

Extend to vision, audio, and video domains:

- **Visual Experiences**: "When identifying faces in low light, enhance contrast before detection"
- **Audio Experiences**: "For speech recognition with background noise, apply spectral subtraction"
- **Video Experiences**: "For action recognition, focus on motion boundaries"

11.5 Lifelong Learning Systems

Training-Free GRPO enables **lifelong learning** without catastrophic forgetting:

- Continuously add experiences as new tasks emerge
- No need to retrain on historical data (model stays frozen)
- Easy rollback if new experiences degrade performance
- Natural curriculum as library grows in sophistication

11.6 Hybrid Approaches

Combine Training-Free GRPO with parameter updates:

- **Phase 1**: Rapid prototyping with TF-GRPO (hours, $18)
- **Phase 2**: Distill experiences into fine-tuned weights (days, $1,000)
- **Benefit**: Fast iteration cycles + eventual parameter efficiency

11.7 Theoretical Extensions

Deeper exploration of Hamiltonian mechanics in LLMs:

- **Symplectic Geometry**: Conservation laws in high-dimensional policy manifolds
- **Action Principles**: Least-action paths in context space optimization
- **Phase Transitions**: Critical points where context expansion becomes counterproductive

12 Conclusion

We have presented **Hamiltonian Large Language Models (HLLM)** with **Training-Free Group Relative Policy Optimization (TF-GRPO)**, a paradigm shift from parameter space to context space learning. By grounding optimization in Hamiltonian mechanics through the conservation law $\Psi \cdot \Theta = \kappa$, we demonstrate that frozen foundation models can achieve state-of-the-art performance by curating semantic experiences in their context windows.

12.1 Key Contributions Revisited

1. **Revolutionary Cost Reduction**: 99.8% cheaper than fine-tuning ($18 vs. $10,000+) with *better* performance (+2.7% on AIME24)
2. **Data Efficiency**: 100× fewer training samples (100 vs. 10,000+)
3. **Transparent AI**: Human-readable experiences replace black-box parameter updates
4. **Superior Transfer**: Maintains performance across domains where fine-tuned models collapse
5. **Decentralized Governance**: DAO-based experience curation with cryptographic verification
6. **Democratized Access**: Open-source implementation in Gym platform, deployed on Zoo Network

12.2 Broader Impact

This work has profound implications for AI accessibility, safety, and governance:

- **Economic**: Makes advanced AI affordable for researchers, educators, and small organizations worldwide
- **Scientific**: Enables rapid experimentation and domain adaptation at 1/500th the cost
- **Social**: Transparent, community-governed AI systems reduce risks of centralized control
- **Technical**: Modular experiences as "software libraries" for LLMs enable new architectures

12.3 The Road Ahead

Training-Free GRPO represents the first step toward a new paradigm: *curated knowledge systems* rather than *trained neural networks*. As foundation models grow more capable, the bottleneck shifts from *what the model knows* to *how to activate the right knowledge at the right time*. Semantic experiences provide a human-readable, governable, and verifiable solution to this challenge.

We envision a future where:

- Communities collaboratively curate domain-specific experience libraries
- Experiences are traded in open marketplaces, creating economic incentives for quality
- AI systems explain their decisions by citing the experiences they applied
- Harmful behaviors are removed by democratic vote, not opaque technical interventions
- Anyone with $20 and 100 examples can adapt frontier models to their needs

This is not merely an optimization technique—it is a fundamental rethinking of how we build, deploy, and govern artificial intelligence. By embracing transparency, decentralization, and community governance, Training-Free GRPO charts a path toward AI systems that are not only more capable and affordable, but also more aligned with human values and democratic principles.

Acknowledgments

This research was conducted by Zoo Labs Foundation, a 501(c)(3) non-profit organization dedicated to democratizing AI. We thank the Hanzo Network team for compute infrastructure, the Lux blockchain team for consensus layer integration, and the global community of contributors to the Gym platform. Special thanks to Tencent's youtu-agent team for pioneering the training-free GRPO algorithm, upon which this work builds.

Funding for this research was provided by community donations to zoo.ngo and KEEPER token holders of the Zoo Network DAO.

References

[1] Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J. and Amodei, D., 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

[2] Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D.D.L., Hendricks, L.A., Welbl, J., Clark, A. and Hennigan, T., 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

[3] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A. and Schulman, J., 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, pp.27730-27744.

[4] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A. and Hassabis, D., 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13), pp.3521-3526.

[5] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V. and Zhou, D., 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, pp.24824-24837.

[6] Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y. and Narasimhan, K., 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

[7] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S., 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, pp.1877-1901.

[8] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Zhang, M., Li, Y., Wu, Y. and Guo, D., 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

[9] Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C. and Chen, C., 2022. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

[10] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O., 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[11] Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S. and Finn, C., 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

[12] Meng, Y., Xie, S.M., Agarwal, A., Zuo, S., Yin, W. and Liu, T., 2024. SimPO: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

[13] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.T., Rocktäschel, T. and Riedel, S., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33, pp.9459-9474.

[14] Li, X.L. and Liang, P., 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

[15] Lester, B., Al-Rfou, R. and Constant, N., 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

[16] Laskin, M., Wang, L., Oh, J., Parisotto, E., Spencer, S., Steigerwald, R., Strouse, D.J., Hansen, S., Filos, A., Brooks, E. and Gazeau, M., 2022. In-context reinforcement learning with algorithm distillation. *arXiv preprint arXiv:2210.14215*.

[17] Greydanus, S., Dzamba, M. and Yosinski, J., 2019. Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32.

[18] Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D. and Ho, S., 2020. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*.

[19] Opper, M. and Archambeau, C., 2009. The variational Gaussian approximation revisited. *Neural Computation*, 21(3), pp.786-792.

[20] McMahan, B., Moore, E., Ramage, D., Hampson, S. and Arcas, B.A.Y., 2017. Communication-efficient learning of deep networks from decentralized data. *Artificial Intelligence and Statistics*, pp.1273-1282.

[21] Salah, K., Rehman, M.H.U., Nizamuddin, N. and Al-Fuqaha, A., 2019. Blockchain for AI: Review and open research challenges. *IEEE Access*, 7, pp.10127-10149.

[22] Ghodsi, Z., Gu, T. and Garg, S., 2017. SafetyNets: Verifiable execution of deep neural networks on an untrusted cloud. *Advances in Neural Information Processing Systems*, 30.

[23] Cover, T.M., 1999. Elements of information theory. *John Wiley & Sons*.

[24] Liu, Y., Zhang, R., Song, J., Guo, D. and Wang, P., 2024. ReTool: Enhancing Mathematical Reasoning via Reinforcement Learning with Tool Integration. *arXiv preprint arXiv:2409.17145*.

[25] Zhou, Y., Muresanu, A.I., Han, Z., Paster, K., Pitis, S., Chan, H. and Ba, J., 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

[26] Tencent Cloud ADP, 2025. Training-Free Group Relative Policy Optimization: A Novel Approach to Efficient LLM Improvement. *arXiv preprint arXiv:2510.08191v1*.

[27] Zoo Labs Foundation, 2025. Gym: Open-Source AI Model Training Platform. *GitHub repository*, `https://github.com/zooai/gym`.

[28] Hanzo Network, 2025. Decentralized Compute Infrastructure for AI Workloads. *Technical Documentation*, `https://hanzo.network`.

[29] Lux Blockchain, 2025. Multi-Consensus Blockchain Architecture with Post-Quantum Cryptography. *Technical Whitepaper*, `https://lux.network`.

[30] Zoo Network, 2025. Decentralized AI/ML Blockchain with Experience-Based Learning. *Technical Documentation*, `https://zoo.ngo`.