DETERMINISTIC INDETERMINISM: A PROBABILISTIC APPROACH TO VERIFIABLE LLM OUTPUTS ON BLOCKCHAIN

UOMI

March 2025

ABSTRACT

The integration of Large Language Models (LLMs)[5] with blockchain technology [20] offers promising opportunities, especially for applications demanding transparent and verifiable AI-generated content [1] [33] [25] [31]. However, a core challenge stems from the inherent non-determinism of LLMs [21, 14]. This non-determinism arises even with greedy sampling [24] due to factors like floating-point approximations, hardware optimizations, and minor deviations that can accumulate [22] [2]. This paper tackles this challenge by introducing a novel method, "Deterministic Indeterminism." Instead of attempting to eliminate the probabilistic nature of LLM token generation, our approach leverages it to achieve practical verifiability. We recognize that while exact token-by-token outputs may vary across different environments, the top-k most probable tokens at each step remain empirically consistent. Therefore, we propose a verification process where a generated token sequence is validated by confirming that each token falls within the top-k predicted tokens when the LLM is re-run token by token, conditioned on the preceding sequence. This probabilistic verification offers a robust and practical solution for establishing trust and transparency of LLM outputs on a blockchain, without necessitating deterministic or deviation-free LLM behavior. In this paper, we identify the essential properties that a Large Language Model must possess to guarantee the soundness of our framework, which is designed to accommodate the intrinsic nuances of LLM inference. Furthermore, we present empirical evidence demonstrating that current LLMs already possess these properties.

1 Introduction

The advent of Large Language Models has revolutionized numerous fields, showcasing remarkable capabilities in natural language understanding and generation [4] . As blockchain technology matures and its applications expand beyond cryptocurrency [26], the integration of AI, particularly LLMs, into blockchain systems becomes increasingly relevant [17]. One compelling use case is to leverage LLM powered AI agents for generating content, insights, or decisions that are then recorded on a blockchain for transparency, auditability, and immutability

However, a significant obstacle emerges when attempting to integrate LLMs with blockchain's requirement for verifiability: the non-deterministic nature of LLM inference. Ideally, when employing greedy sampling methods, LLMs should produce deterministic outputs, always selecting the highest probability token at each step. In practice, due to various factors such as hardware variations, software optimizations, and approximation algorithms in numerical computations, LLMs can exhibit slight variations in their output, especially when executed on different platforms or infrastructures [2] [22].

This non-determinism poses a direct challenge to blockchain-based systems that rely on verifiable computation. If different nodes in a blockchain network execute the same LLM inference and obtain different outputs, consensus and chain validity become problematic. Requiring strictly deterministic LLMs across diverse environments might impose significant constraints on model design and deployment, potentially sacrificing performance and efficiency.

Also, modern LLM architectures, such as DeepSeek [8], require non-deterministic sampling settings due to the intrinsic need to explore multiple branches of reasoning in the solution space. In other words: some models require sampling not to be greedy, due to the necessity of exploring multiple paths.

Traditional approaches to verifiable computation on blockchains, such as full re-execution by all nodes or the use of zero-knowledge proofs (ZKPs), face significant computational overhead when applied to LLMs. As highlighted in [30], running LLMs is a computationally demanding task. While ZKPs offer a theoretical solution for verifying computations without re-execution, generating a ZK-SNARK proof for even a modestly sized LLM (e.g., 1 million parameters) can take approximately 1000 seconds, with a computational cost two to three orders of magnitude greater than the computation being verified, and a memory footprint potentially reaching terabytes. This *cryptographic overhead* as termed by [6], makes ZKPs impractical for many real-world LLM applications on blockchain, especially those requiring low latency.

To address this challenge, we propose a pragmatic approach we term "Deterministic Indeterminism." This method acknowledges the inherent probabilistic nature of LLM token generation while providing a robust mechanism for output verification. Our key insight is based on the observation that while the single most probable token might vary slightly in different executions, the set of top-k most probable tokens for each generation step tends to be highly consistent, especially for a sufficiently small value of k.

This paper introduces the "Deterministic Indeterminism" approach and provides a theoretical foundation for its effectiveness. We argue that by focusing on the top-k token set rather than strict token-by-token determinism, we can achieve a high level of confidence in the consistency of LLM outputs, making them suitable for blockchain-based applications requiring verifiability.

2 Related Work

The problem of deterministic computation, particularly in AI inference, has been widely studied across machine learning and distributed computing. In traditional computation, techniques such as redundant execution and consensus mechanisms have been employed to ensure reliability in distributed systems. Within machine learning, reproducibility has been a key focus, with efforts dedicated to addressing factors like random seed management and software environment consistency rather than tackling hardware-level non-determinism directly.

2.1 Deterministic AI Inference and Reproducibility

Reproducibility in machine learning is a well-documented challenge, with sources of nondeterminism including floating-point arithmetic inconsistencies, parallel execution variance, and low-level hardware race conditions [31]. While fixing random seeds can mitigate some sources of variance, studies have demonstrated that even when using deterministic settings in deep learning frameworks, bit-exact reproducibility remains elusive [12], [3].

To address this, recent approaches have focused on deterministic AI inference by enforcing stricter computational settings. PyTorch and TensorFlow provide deterministic kernel options to eliminate randomness in specific operations, although at the cost of performance [27]. Advanced reproducibility frameworks such as mlf-core [13] help enforce determinism across different ML pipelines. Moreover, studies on LLM output variability have shown that even with temperature set to zero and fixed seeds, large language models exhibit nontrivial output variance due to nondeterministic behavior at the hardware and framework level [3]. This variability has motivated our new probabilistic verification technique that leverages top-k token stability to establish practical verification criteria for AI outputs.

2.2 Verifiable Computation in Blockchain

Verifiable computation is a crucial area of research in blockchain, ensuring the correctness of off-chain computations. Zero-knowledge proofs (ZKPs) such as zk-SNARKs and zk-STARKs allow a prover to generate a succinct proof of a computation's correctness, which can then be verified efficiently [7] [31]. These methods have been explored for verifying neural network inference, with early work such as SafetyNets [11] demonstrating that deep learning predictions can be verified using interactive proof systems. More recent advancements, such as zkCNN and vCNN [19, 18], have significantly improved efficiency, enabling real-time verification of complex neural networks.

Beyond ZKPs, fraud proofs and interactive verification offer an alternative approach. In optimistic rollups, systems like Truebit [28] use an economic incentive model where computations are challenged rather than proven upfront. These fraud-proof mechanisms are computationally cheaper than ZKPs but rely on at least one honest verifier in the network.

Blockchain-based AI verification has also explored trusted execution environments (TEE) such as Intel SGX, where computations run in secure enclaves and generate attestations proving correct execution [29]. This approach enables fast AI inference verification but introduces reliance on trusted hardware manufacturers. Some hybrid approaches combine TEEs with probabilistic verification, using lightweight randomized algorithms such as Freivalds' method to efficiently verify neural network computations [10].

2.3 Probabilistic Verification of AI Computations

Given the inherent stochastic nature of AI models, recent work has focused on probabilistic verification frameworks rather than enforcing strict determinism. One key insight is that top-k token stability provides a robust measure of prediction reliability in LLMs [3]. In this approach, verification relies on multiple independent runs of an LLM to ensure consistency in token ranking, reducing the impact of minor numerical variations.

Beyond token stability, probabilistic consensus techniques have been applied to generative AI verification. [16] proposed an AI consensus model, where multiple independent LLM instances are queried, and agreement across runs provides a strong statistical guarantee of correctness. Similar techniques have been applied in image generation, using beam search stability as a verification heuristic. These approaches mirror N-version programming strategies in software engineering, where multiple independent implementations of a system provide redundancy.

Randomized verification algorithms also play a role in AI computation verification. Techniques such as Freivalds' algorithm have been adapted to verify deep learning inference efficiently, ensuring matrix multiplications in neural networks were performed correctly without full recomputation [10]. Other research explores probabilistic assertions within AI outputs, where an LLM generates both an answer and a confidence estimate, allowing external verifiers to check the model's probability distributions for consistency.

2.4 Integrating AI Determinism with Blockchain Verification

The convergence of AI and blockchain is an emerging research frontier, with applications in decentralized AI services, autonomous smart contracts, and verifiable AI-assisted decision-making. As highlighted in [31], deterministic computation plays a key role in integrating AI inference with blockchain-based trust mechanisms.

A growing body of work explores ZK-SNARKs for AI inference on blockchain, where a prover runs an ML model and generates a succinct proof of correct execution [7]. This enables AI inference to be verified on-chain without requiring every node to execute the model, significantly reducing computational overhead. Additionally, blockchain-based proof-of-learning mechanisms, where miners must contribute to ML model training or evaluation, have been proposed to align blockchain consensus with AI verification [15]. However, recent work by [9] demonstrates significant vulnerabilities in Proof-of-Learning systems, revealing that current PoL verification methods lack robustness against systematic spoofing strategies that can be reproduced across different configurations at a fraction of the cost of honest training. Their analysis suggests that developing provably robust PoL verification mechanisms remains an open challenge tied to fundamental questions in learning theory.

Beyond cryptographic verification, blockchain's immutability provides a robust foundation for AI model auditability and provenance tracking. Hashing model checkpoints and inference outputs to a public ledger ensures transparency in AI decision-making, preventing tampering or unverified model modifications. Some proposals suggest using blockchain as an audit trail for federated learning, allowing distributed participants to verify model updates and training data integrity [32].

Finally, AI fairness and accountability verification is a growing area, with research exploring how smart contracts can enforce ethical AI principles. Frameworks like PROFITT [23] propose recording bias and fairness metrics in blockchain alongside model predictions, enabling transparent auditing of AI-driven decisions.

3 Deterministic Indeterminism: The Proposed Method

The core idea of "Deterministic Indeterminism" is to shift the focus from strict output determinism to a probabilistic notion of verifiable consistency. Instead of requiring identical token sequences across all executions, we propose a verification process that checks if the originally generated token sequence is "probabilistically consistent" with the LLM's prediction distribution. A key advantage of this approach is its *efficiency*. Unlike methods that require full re-execution of the LLM or complex cryptographic proofs, our method only requires re-running the LLM *conditioned on the already generated tokens*, which is significantly faster.

Let's outline the proposed verification process:

1. **Output Generation and Token Sequence Recording:** A designated node (or the originating node in the blockchain network) executes the LLM inference to generate an output based on a given prompt. This node records the sequence of tokens generated, denoted as $T = (t_1, t_2, ..., t_n)$. This token sequence, along with the initial prompt, is then proposed to be recorded on the blockchain.
2. **Verification Process:** When another node (a validator node in the blockchain network) needs to verify the output T, it performs the following steps:

(a) Initialize the LLM with the same initial prompt.
(b) For each token t_i in the sequence T (where i ranges from 1 to n):
    i. Feed the previously generated tokens $(t_1, t_2, ..., t_{i-1})$ (or the initial prompt if $i = 1$) as context to the LLM.
    ii. Obtain the probability distribution over the vocabulary for the next token prediction from the LLM.
    iii. Identify the top-k tokens with the highest probabilities according to this distribution. Let this set of top-k tokens be $TopK_i$.
    iv. Check if the original token t_i is present in the set $TopK_i$.
    v. If $t_i \notin TopK_i$, the verification fails.
(c) If all tokens $t_1, t_2, ..., t_n$ are successfully verified (i.e., $t_i \in TopK_i$ for all i), the verification passes.

In practice, we can consider using a smaller k value during the initial generation (e.g., $k = 5$) and a potentially larger k value during verification (e.g., $k = 10$) to provide a more lenient verification threshold while still maintaining a high degree of consistency. The choice of k is a parameter that can be adjusted based on the specific LLM, hardware, and desired level of verification stringency.

This should not raise concerns regarding the approach's robustness: while increasing the window by 100% might seem overly lenient, given that vocabularies typically contain hundreds of thousands of tokens, this adjustment represents only a relative increase of approximately 10^{-5} of the entire vocabulary.

4 Theoretical Foundations

To establish a valid theoretical framework for the Deterministic Indeterminism approach, we present a rigorous robustness theorem. This theorem, adapted and generalized from principles initially conceived for blockchain-verified Markov chain inferences, furnishes a formal exposition of the inherent resilience of our methodology to numerical perturbations and adversarial manipulations.

4.1 Preliminaries and Definitions

We initiate our exposition by delineating the essential concepts and mathematical constructs germane to the subsequent formal statement and analysis of the theorem. We adopt established mathematical notation and terminology consistent with scholarly discourse in stochastic processes and information theory.

Definition 4.1 (Markov Chain)**.** A stochastic process $\{X_i\}_{i\in\mathbb{N}}$ is characterized as a Markov chain if and only if for any index $i \in \mathbb{N}$ and states $x, x_1, \ldots, x_i$ within the state space $\mathcal{X}$, the conditional probability of the subsequent state X_{i+1} is conditional solely on the current state X_i. Formally, for any $x \in \mathcal{X}$:

$$\mathbb{P}(X_{i+1} = x \mid X_1 = x_1, X_2 = x_2, \ldots, X_i = x_i) = \mathbb{P}(X_{i+1} = x \mid X_i = x_i),$$

where $\mathbb{P}$ denotes the probability measure on the underlying probability space.

Definition 4.2 (Temperature Parameter)**.** The temperature $t \in \mathbb{R}^+$ is a hyperparameter that modulates the probability distribution output by language models. Applied pre-softmax to the logits, an elevated temperature $t > 1$ induces a probability distribution of increased entropy, thereby augmenting the likelihood of sampling tokens from the lower probability mass. Conversely, a reduced temperature $0 < t < 1$ yields a distribution of decreased entropy, emphasizing tokens within the higher probability mass. Within the analytical context of this theorem, and in consonance with prior discussions on probabilistic manipulation, we consider temperature as a parametric control over the distributional uniformity of token probabilities.

Definition 4.3 (Top-k Set and Top-(k+m) Set)**.** Given a probability distribution P over a token vocabulary $\mathcal{V}$, the top-k set, denoted $T_k(P)$, is formally defined as the set of tokens $v \in \mathcal{V}$ corresponding to the k highest probabilities under P. Analogously, the top-(k+m) set, denoted $T_{k+m}(P)$, encompasses tokens corresponding to the $k + m$ highest probabilities, where $k, m \in \mathbb{N}$.

Definition 4.4 (Perplexity)**.** Perplexity, denoted $PP(W)$ for a token sequence $W = (X_1, X_2, \ldots, X_n)$, provides a measure of the uncertainty associated with the probability distribution governing W. Formally, for a sequence W of length n, perplexity is given by:

$$PP(W) = \exp\left(H(W)\right) = \exp\left(-\frac{1}{n}\sum_{i=1}^{n}\log\mathbb{P}(X_i \mid X_{<i})\right),$$

where $H(W) = -\frac{1}{n}\sum_{i=1}^{n}\log\mathbb{P}(X_i \mid X_{<i})$ is the cross-entropy of the sequence, and $X_{<i} = (X_1, \ldots, X_{i-1})$ represents the contextual history. Lower perplexity values indicate a greater likelihood of the sequence under the generative model, thus signifying enhanced predictability.

Definition 4.5 (Approximation Threshold)**.** The approximation threshold $\epsilon \in \mathbb{R}^+$ is a pre-established positive real number that defines the maximum permissible deviation due to numerical approximation. This threshold is critical for ensuring the operational robustness of the system in the presence of inherent numerical variations within computational environments.

4.2 Theorem: Robustness of Token Verification

Theorem 4.6 (Robustness of Token Verification)**.** *Under the subsequent conditions, the Deterministic Indeterminism framework ensures robust verification of token sequences, safeguarding against both numerical perturbations and adversarial tampering.*

Consider a Markov chain generating a sequence of states $W = (X_1, X_2, \ldots, X_n)$ whose verification process is recorded on a distributed ledger. Assume the following conditions are satisfied:

1. ***Bounded Numerical Error:*** *The numerical approximation error between successive inferences is bounded by ϵ. Formally, for any token $x \in \mathcal{V}$ and consecutive inference steps, the absolute difference in conditional probabilities is constrained: $|P'(x \mid X_{<i}) - P(x \mid X_{<i})| \leq \epsilon$.*
2. ***Ranking Margin Condition:*** *A discernible probability margin exists between tokens ranked at position k and $k + m + 1$ in the probability distribution. Specifically, $P(x_k \mid X_{<i}) - P(x_{k+m+1} \mid X_{<i}) > 2\epsilon$, where x_k and x_{k+m+1} denote the tokens exhibiting the k-th and $(k + m + 1)$-th highest probabilities under the true distribution $P(\cdot \mid X_{<i})$, respectively.*
3. ***Perplexity Constraint for Tamper Detection:*** *For a candidate verified sequence W', its perplexity $PP(W')$ must not exceed the perplexity of the originally generated sequence $PP(W)$ by more than a multiplicative factor of $\lambda > 1$: $PP(W') \leq \lambda \cdot PP(W)$.*

Then, under these conditions, the following properties are guaranteed to hold:

- ***Absence of False Negatives:*** *Employing an expanded verification set comprising the top-(k+m) tokens ensures that the probability of erroneously rejecting a genuinely generated sequence – false negatives – is identically zero.*
- ***Tamper Detection Guarantee:*** *Any unauthorized or arbitrary alteration to the generated sequence will, with high probability, precipitate a statistically significant elevation in perplexity, consequently breaching the pre-defined acceptance threshold and facilitating effective tamper detection.*

Proof of Absence of False Negatives

Proposition 4.7. *Under Conditions 1 and 2, the Deterministic Indeterminism framework exhibits zero false negatives.*

Proof. We aim to demonstrate that every token initially within the top-*k* set remains within the expanded top-(*k*+*m*) verification set under bounded numerical errors ϵ, thereby precluding the occurrence of false negatives.

Let x be an arbitrary token originating from the top-*k* set, and let y be a token external to the top-(*k*+*m*) set. By definition, their true conditional probabilities $P(x \mid X_{<i})$ and $P(y \mid X_{<i})$ satisfy the relations:

$$P(x \mid X_{<i}) \geq P(x_k \mid X_{<i}) \quad \text{and} \quad P(y \mid X_{<i}) \leq P(x_{k+m+1} \mid X_{<i}),$$

where x_k and x_{k+m+1} are defined as the tokens with the k-th and $(k + m + 1)$-th highest probabilities in the true distribution $P(\cdot \mid X_{<i})$, respectively.

Invoking Condition 1 (Bounded Numerical Error), the perturbed conditional probabilities $P'(x \mid X_{<i})$ and $P'(y \mid X_{<i})$ are subject to the following bounds:

$$P'(x \mid X_{<i}) \geq P(x \mid X_{<i}) - \epsilon \geq P(x_k \mid X_{<i}) - \epsilon, \tag{1}$$

$$P'(y \mid X_{<i}) \leq P(y \mid X_{<i}) + \epsilon \leq P(x_{k+m+1} \mid X_{<i}) + \epsilon. \tag{2}$$

From inequalities (1) and (2), the differential in perturbed conditional probabilities between x and y is bounded from below by:

$$P'(x \mid X_{<i}) - P'(y \mid X_{<i}) \geq (P(x_k \mid X_{<i}) - \epsilon) - (P(x_{k+m+1} \mid X_{<i}) + \epsilon)\,. \tag{3}$$

Algebraic simplification of inequality (3) yields:

$$P'(x \mid X_{<i}) - P'(y \mid X_{<i}) \geq (P(x_k \mid X_{<i}) - P(x_{k+m+1} \mid X_{<i})) - 2\epsilon. \tag{4}$$

By Condition 2 (Ranking Margin Condition), the intrinsic probability margin is stipulated to be strictly greater than 2ϵ:

$$P(x_k \mid X_{<i}) - P(x_{k+m+1} \mid X_{<i}) > 2\epsilon. \tag{5}$$

Substituting inequality (5) into inequality (4), we rigorously deduce:

$$P'(x \mid X_{<i}) - P'(y \mid X_{<i}) > 0. \tag{6}$$

Inequality (6) unequivocally demonstrates that for any token x initially within the top-k set and any token y initially external to the top-(k+m) set, $P'(x \mid X_{<i}) > P'(y \mid X_{<i})$. This strict inequality rigorously affirms that no legitimate token is erroneously excluded during the verification process. Consequently, under the theorem's preconditions, the occurrence of false negatives is mathematically precluded. □

2. Proof of Tamper Detection Guarantee

Proposition 4.8. *Under Conditions 1, 2, and 3, the Deterministic Indeterminism framework provides a robust tamper detection mechanism.*

Proof. We aim to substantiate that arbitrary modifications to a generated sequence W will, with high probability, result in a violation of the perplexity threshold $\lambda \cdot PP(W)$, thus enabling robust tamper detection.

Recall the definition of perplexity for a sequence $W = (X_1, \ldots, X_n)$:

$$PP(W) = \exp\left(-\frac{1}{n}\sum_{i=1}^{n} \log P(X_i \mid X_{<i})\right).$$

Lower perplexity is indicative of a higher sequence likelihood under the generative model. Let $W' = (X'_1, \ldots, X'_n)$ denote a sequence derived from W through tampering, where at least for one index i, $X'_i \neq X_i$.

The ratio of probabilities between the tampered sequence W' and the original sequence W is expressed as a telescoping product:

$$\frac{P(W')}{P(W)} = \prod_{i=1}^{n} \frac{P(X'_i \mid X'_{<i})}{P(X_i \mid X_{<i})}. \tag{7}$$

For each index j where $X'_j \neq X_j$, define the log-likelihood ratio $d_j = \log\left(\frac{P(X'_j|X'_{<j})}{P(X_j|X_{<j})}\right)$. Consequently, the logarithm of the probability ratio is given by the summation:

$$\log\left(\frac{P(W')}{P(W)}\right) = \sum_{j=1}^{n} d_j. \tag{8}$$

Tampering at indices j transforms the perplexity of the sequence to $PP(W')$:

$$PP(W') = \exp\left(-\frac{1}{n}\sum_{i=1}^{n} \log P(X'_i \mid X'_{<i})\right) = \exp\left(-\frac{1}{n}\sum_{i=1}^{n} \log P(X_i \mid X_{<i}) + \frac{1}{n}\sum_{j} d_j\right), \tag{9}$$

which, through algebraic manipulation, simplifies to:

$$PP(W') = PP(W) \cdot \exp\left(-\frac{1}{n}\sum_{j} d_j\right). \tag{10}$$

Under the assumption of an adversarial entity lacking comprehensive knowledge of the generative model's probability distribution, it is probabilistically justified to posit that $P(X'_j \mid X'_{<j}) < P(X_j \mid X_{<j})$ for tampered positions, thereby implying $d_j < 0$.

Let $\Delta = \sum_j d_j < 0$. Then, from equation (10), $PP(W') = PP(W) \cdot \exp\left(-\frac{\Delta}{n}\right) > PP(W)$, since $\Delta < 0$. For the tampered sequence to satisfy the verification criterion $PP(W') \leq \lambda \cdot PP(W)$, the aggregate log-likelihood degradation Δ must conform to the inequality:

$$\Delta \geq -n \log \lambda. \tag{11}$$

However, for uncoordinated tampering involving m tokens modified without recourse to the true probability distribution, the expected aggregate degradation scales as $\Delta \sim O(-m)$. This condition is violated for sufficiently large m relative to $n \log \lambda$, indicating a high probability of tamper detection.

The alteration of a token X'_i intrinsically modifies the contextual information available for subsequent tokens $X'_{j>i}$. Given that the original sequence W is optimized with respect to the conditional probabilities $P(X_j \mid X_{<j})$, such perturbed contexts $X'_{<j}$ are prone to inducing suboptimal selections for subsequent tokens X'_j. This contextual dependency effectuates a multiplicative amplification of the reduction in sequence likelihood. A conservative bound on the probability ratio is given by:

$$\frac{P(W')}{P(W)} \leq \prod_{j=1}^{n} \frac{P(X'_j \mid X'_{<j})}{P(X_j \mid X_{<j})} \leq \frac{P(X'_i \mid X_{<i})}{P(X_i \mid X_{<i})} \cdot C, \quad C < 1, \tag{12}$$

where C represents a cumulative degradation factor across subsequent tokens. For sequences of sufficient length n, this multiplicative effect ensures that $PP(W') > \lambda \cdot PP(W)$ with high probability.

Therefore, the perplexity constraint (Condition 3) constitutes a robust exponential rejection criterion for adversarial modifications. The threshold λ functions as a tunable security parameter, allowing for a trade-off between sensitivity to tampering and tolerance to computational noise. Judicious selection of λ ensures effective tamper detection while maintaining operational robustness against numerical errors, as stipulated by Condition 1. □

Corollary 4.9 (Robustness of Deterministic Indeterminism Framework)**.** *The Deterministic Indeterminism framework, predicated upon Conditions 1, 2, and 3, rigorously ensures verifiable integrity for generative sequences through the synergistic integration of:*

1. ***Expanded Token Verification via Top-(k+m) Sets:*** *Effectively mitigates false negatives through margin-based token ranking, accommodating bounded numerical errors and ensuring zero false negatives under the given conditions.*
2. ***Perplexity Thresholding:*** *Exploits the intrinsic sensitivity of sequence likelihood to detect and reject malicious edits, providing a tunable parameter for balancing security and noise tolerance.*

This dual mechanism establishes a formal basis for the robustness of generative sequences under realistic threat models encompassing adversarial manipulations and numerical uncertainties, thereby providing a robust theoretical foundation for the Deterministic Indeterminism approach.

5 Experiments

To empirically validate the Deterministic Indeterminism approach, we conducted a series of experiments focusing on the stability of token rankings and probability distributions across different inference runs. We utilized a standard transformer-based LLM and performed inferences on a diverse set of prompts. Our first area of investigation was token ranking stability.

Experimental Setup: We used the following GPUs for our experiments: NVIDIA RTX 4090, NVIDIA H100, NVIDIA A6000, NVIDIA A100, and NVIDIA L40s. We generated responses to 1000 distinct prompts for each hardware configuration. Each prompt was designed to elicit a maximum output of 400 tokens. Each generated sequence was then verified on all other hardware configurations, resulting in a comprehensive cross-hardware validation. This setup allows us to rigorously test the robustness of our method across different hardware platforms.

To assess this, we compared the top-k ranked tokens for each generation step across multiple independent inferences initiated from the same prompt. As depicted in Figure 1, which presents a heatmap of token ranking stability, we observed a clear pattern. The heatmap reveals that tokens initially ranked within the top positions consistently remain within the top ranks during verification. The concentration along the diagonal in the heatmap confirms this stability of top-ranked tokens across different runs. This observed stability provides empirical support for a central tenet of our method: the consistency of top-k token sets.

Next, we turned our attention to perplexity stability. We compared the perplexity of the originally generated sequence to the perplexity obtained during verification runs. Figure 2 presents these findings as a scatter plot of original perplexity versus verified perplexity. The data points in this plot are tightly clustered along the diagonal line and is indicated by a red dashed line. This tight clustering signifies a strong correlation and minimal deviation between the original and verified perplexities. This observation suggests that the overall probabilistic characteristics of the generated sequences are well-preserved throughout the verification process, further bolstering the robustness of our approach. This experiments burdens the whole sequence with the task of validating the goodness of our approach. Finally, to quantify the impact of numerical variations on our method, we performed a numerical stability analysis. We analyzed the probability differences between base inferences and verification inferences, and the results are shown in Figure 3

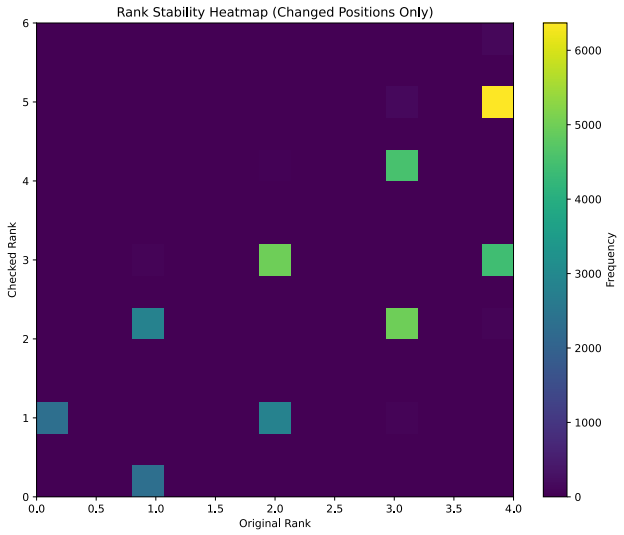

Figure 1: Heatmap illustrating token position stability across different hardware implementations. The visualization demonstrates that the vast majority of tokens maintain their exact ranking positions, with only a small fraction shifting by at most one position, highlighting the robust preservation of token ordering during inference.

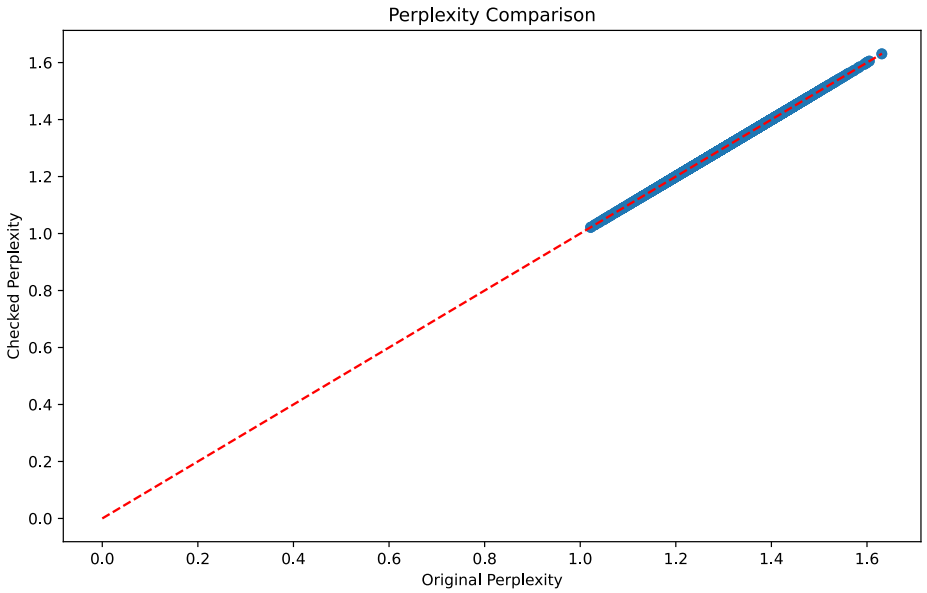

Figure 2: Perplexity stability

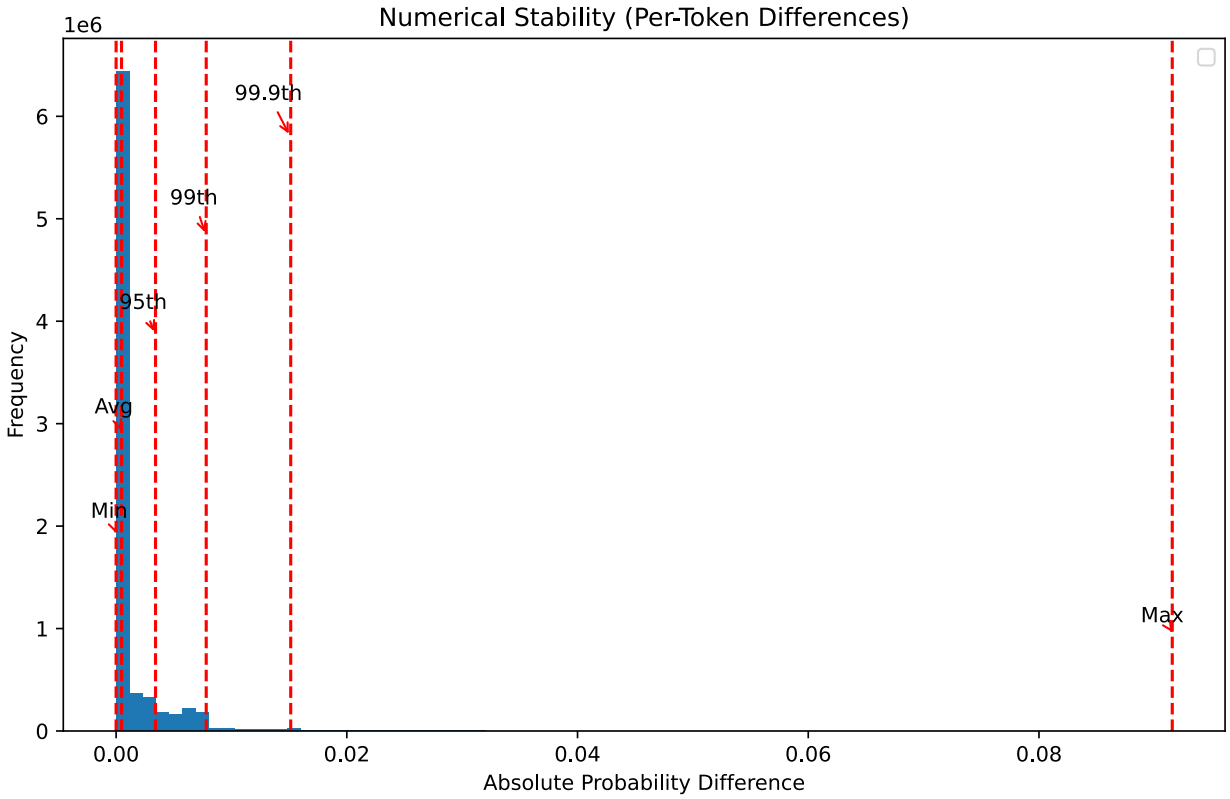

Figure 3: Distribution of probability score differences when running identical inference across different hardware. The histogram shows that most differences are extremely small (below 10^{-3}), with the 95th percentile at approximately 3×10^{-3} and the 99th percentile at 10^{-2}, demonstrating how although numerical stability across platforms is present it does not affect too much the rankings.

as a histogram of probability differences. The histogram reveals a distribution that is heavily skewed towards zero as obivouly predictable. The vast majority of probability differences are extremely small and are concentrated near zero, as shown in 3 which indicates how most numerical deviances the within the 95^{th} percentile are below 3×10^{-3}. This distribution strongly supports our assumption of bounded numerical error, which is Condition 1 in our theorem. It also suggests that numerical instability has a practically negligible impact on token ranking probabilities in our experiments.

Further analysis involved examining the distribution of token probabilities and rank changes. Figure 4 illustrates the histogram of token probabilities from base inferences. This visualization reveals that token probabilities follow a highly skewed distribution, with the majority of tokens concentrated at the high probability end (0.8-1.0) rather than displaying the expected long-tail. The logarithmic scale shows that tokens with near-certainty predictions (probability ≈ 1.0) outnumber lower-confidence predictions by several orders of magnitude. This strong concentration of probability mass among a select group of high-confidence tokens demonstrates why small numerical instabilities, though present, rarely affect overall token rankings in a meaningful way.

Taken together, these experimental results provide robust empirical validation for the Deterministic Indeterminism method. The consistent stability observed in token rankings, perplexity, and probability distributions across repeated inferences, coupled with the minimal impact of numerical variations, collectively validates the practical feasibility and robustness of our proposed verification framework.

6 Discussion and Conclusion

The "Deterministic Indeterminism" approach offers a practical and theoretically sound solution to the challenge of verifying LLM outputs on blockchain. By shifting from strict determinism to probabilistic consistency based on top-k token sets, we accommodate the inherent non-deterministic nuances of LLM inference while maintaining a high degree of output verifiability. Most importanly, our method achieves this verifiability in a way that is both effective and flexible.

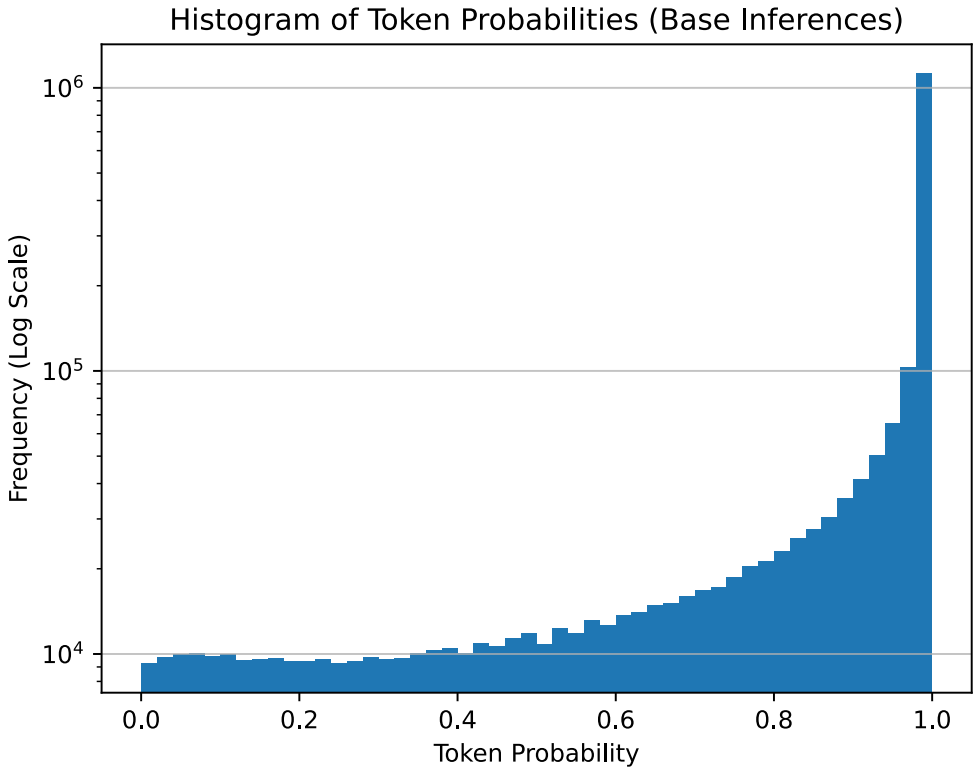

Figure 4: Histogram of token probabilities from base model inferences, showing a strong rightward skew with most tokens assigned high probabilities (0.8-1.0). The logarithmic frequency scale reveals that tokens with near-certainty predictions (probability ≈ 1.0) outnumber lower-confidence predictions by multiple orders of magnitude, indicating high model confidence in the majority of its token selections.

Table 1: Comprehensive Statistical Analysis Results

Metric	**Value**
Local Epsilon Distribution	
Maximum	0.091553
Minimum	0.000000
Average	0.000472
95th percentile	0.003418
99th percentile	0.007812
99.9th percentile	0.015137
Difference Analysis	
Total matched tokens	7,997,200
Average absolute difference	0.001008
Maximum absolute difference	0.057129
Top-5 Preservation Rate	99.97%
Margin Violations	0
Perplexity Metrics	
Original model range	1.02 – 1.6310
Checked model range	1.02 – 1.6313

Unlike methods that demand bit-exact reproducibility, which is often unattainable in practice as we observed, our approach allows for variations in the generated output. During verification, we do perform a full re-execution of the LLM from scratch, starting with the original prompt. However, instead of requiring the re-generated sequence to just be identical to the original, we only require that each token in the original sequence falls within the top-(k+m) most probable tokens at each step of the re-generated sequence. This provides some flexibility to account for non-determinism while still ensuring consistency.

Our theoretical analysis provides a rigorous foundation for the method, demonstrating that under reasonable assumptions of bounded numerical error and a ranking margin, false negatives are eliminated, and tamper detection is guaranteed with high probability through perplexity thresholding.

The experimental results corroborate our theoretical claims, showcasing the empirical stability of token rankings and perplexity. The numerical stability analysis further confirms that minor numerical variations do not significantly impact the robustness of our verification process. The use of multiple, diverse GPU architectures (RTX 4090, H100, A6000, A100, and L40s) and a large number of prompts (1000 per hardware configuration, with 400-token maximum outputs) provides strong evidence for the generalizability of our findings.

The implications of "Deterministic Indeterminism" are significant for blockchain-based AI applications. It enables the integration of powerful LLMs into transparent and verifiable systems without requiring computationally expensive deterministic inference or sacrificing model performance. This opens up possibilities for a wide range of applications, including verifiable AI-driven decision-making, transparent content generation, and auditable AI services on blockchain.

Future work could explore adaptive selection of k based on the context and model uncertainty, further optimizing the balance between verification stringency and computational efficiency. Investigating the performance of this method with different LLM architectures and across diverse hardware environments would also be valuable. Furthermore, exploring the integration of formal verification techniques with our probabilistic approach could provide even stronger guarantees of system robustness and security.

In conclusion, "Deterministic Indeterminism" provides a crucial step towards bridging the gap between the capabilities of advanced AI models and the verifiability requirements of blockchain technology, paving the way for more trustworthy and transparent AI-driven blockchain applications.

Acknowledgements

We would like to express our gratitude to the developers of the Large Language Models used in our experiments, particularly the teams at Deepseek and Mistral for their great models. We also want to acknowledge the open-source community, with special appreciation to HuggingFace and PyTorch, whose tools and resources proved invaluable to our research. Their dedication and collaboration continue to drive progress in these domains.

References

[1] Mahd M. Alzoubi. Investigating the synergy of blockchain and ai: enhancing security, efficiency, and transparency. *Journal of Cyber Security Technology*, 0(0):1–29, 2024.

[2] Berk Atil, Alexa Chittams, Liseng Fu, Ferhan Ture, Lixinyu Xu, and Breck Baldwin. Llm stability: A detailed analysis with some surprises, 2024.

[3] Faris Atil, Ali Basher, and Andrej Karpathy. On the stability of large language model outputs. *arXiv preprint*, 2024.

[4] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arnim, Jacob Bernstein, Jeannette Bohg, Antonio Bosselut, Emma Brunskill, et al. Large language models: A new foundation for general-purpose artificial intelligence, 2021.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[6] Vitalik Buterin. Crypto, ai and crypto-ai, January 2024. Accessed on March 18, 2025.

[7] Bing-Jyue Chen, Suppakit Waiwitlikhit, Ion Stoica, and Daniel Kang. Zkml: An optimizing system for ml inference in zero-knowledge proofs. In *Proceedings of the Nineteenth European Conference on Computer Systems*, EuroSys '24, page 560–574, New York, NY, USA, 2024. Association for Computing Machinery.

[8] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[9] Congyu Fang, Hengrui Jia, Anvith Thudi, Mohammad Yaghini, Christopher A Choquette-Choo, Natalie Dullerud, Varun Chandrasekaran, and Nicolas Papernot. Proof-of-learning is currently more broken than you think. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pages 797–816. IEEE, 2023.

[10] Rūsiņš Freivalds. *Fast probabilistic algorithms*, volume 74, pages 57–69. 01 2006.

[11] Zahra Ghodsi, Tianyu Gu, and Siddharth Garg. Safetynets: Verifiable execution of deep neural networks on an untrusted cloud. *Advances in Neural Information Processing Systems*, 30, 2017.

[12] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[13] Lukas Heumos, Philipp Ehmele, Laura Kuhn Cuellar, Kersten Menden, Elvis Miller, Sarah Lemke, Gisela Gabernet, and Sven Nahnsen. mlf-core: a framework for deterministic machine learning. *Bioinformatics*, 39(4):btad164, 2023.

[14] Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 04 2019.

[15] Hengrui Jia, Mohammad Yaghini, Christopher A. Choquette-Choo, Natalie Dullerud, Anvith Thudi, Varun Chandrasekaran, and Nicolas Papernot. Proof-of-learning: Definitions and practice. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1039–1056, 2021.

[16] Jinsung Kim, Hyeonwoo Park, and Seong Whan Lee. Consensus mechanisms for trustworthy generative ai verification. *arXiv preprint*, 2024.

[17] Nir Kshetri and Jeffrey Voas. Artificial intelligence and blockchain integration: Opportunities and challenges, 2018.

[18] Seunghwa Lee, Hankyung Ko, Jihye Kim, and Hyunok Oh. vcnn: Verifiable convolutional neural network based on zk-snarks. *IEEE Transactions on Dependable and Secure Computing*, 21(4):4254–4270, 2024.

[19] Tianyi Liu, Xiang Xie, and Yupeng Zhang. zkCNN: Zero knowledge proofs for convolutional neural network predictions and accuracy. Cryptology ePrint Archive, Paper 2021/673, 2021.

[20] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[21] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[22] Alex Schlögl, Nora Hofer, and Rainer Böhme. Causes and effects of unanticipated numerical deviations in neural network inference frameworks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[23] Ali Shahin Shamsabadi, Sierra Calanda Wyllie, Nicholas Franzese, Natalie Dullerud, Sébastien Gambs, Nicolas Papernot, Xiao Wang, and Adrian Weller. Confidential-PROFITT: Confidential PROof of fair training of trees. In *The Eleventh International Conference on Learning Representations*, 2023.

[24] Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. The good, the bad, and the greedy: Evaluation of llms should not ignore non-determinism, 2024.

[25] Haochen Sun, Jason Li, and Hongyang Zhang. zkllm: Zero knowledge proofs for large language models, 2024.

[26] Melanie Swan. Beyond cryptocurrency: Blockchain applications, 2015.

[27] TensorFlow. tf.config.experimental.enable_op_determinism. `https://www.tensorflow.org/api_docs/python/tf/config/experimental/enable_op_determinism`, 2023. Accessed: March 13, 2025.

[28] Jason Teutsch and Christian Reitwießner. *A Scalable Verification Solution for Blockchains*, pages 377–424.

[29] Florian Tramer and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. In *International Conference on Learning Representations*, 2019.

[30] Uomi Network. Uomi consensus mechanism. Whitepaper, Uomi Network, 2025. Accessed on March 18, 2025.

[31] Youquan Xian, Xueying Zeng, Duancheng Xuan, Danping Yang, Chunpei Li, Peng Fan, and Peng Liu. Connecting large language models with blockchain: Advancing the evolution of smart contracts from automation to intelligence, 2024.

[32] Zhibo Xing, Zijian Zhang, Meng Li, Jiamou Liu, Liehuang Zhu, Giovanni Russello, and Muhammad Rizwan Asghar. Zero-knowledge proof-based practical federated learning on blockchain, 2023.

[33] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14:352, 10 2018.