The Chronicles of Foundation AI for Forensics of Multi-Agent Provenance

Ching-Chun Chang and Isao Echizen

Abstract—Provenance is the chronology of things, resonating with the fundamental pursuit to uncover origins, trace connections, and situate entities within the flow of space and time. As artificial intelligence advances towards autonomous agents capable of interactive collaboration on complex tasks, the provenance of generated content becomes entangled in the interplay of collective creation, where contributions are continuously revised, extended or overwritten. In a multi-agent generative chain, content undergoes successive transformations, often leaving little, if any, trace of prior contributions. In this study, we investigates the problem of tracking multi-agent provenance across the temporal dimension of generation. We propose a chronological system for post hoc attribution of generative history from content alone, without reliance on internal memory states or external meta-information. At its core lies the notion of symbolic chronicles, representing signed and time-stamped records, in a form analogous to the chain of custody in forensic science. The system operates through a feedback loop, whereby each generative timestep updates the chronicle of prior interactions and synchronises it with the synthetic content in the very act of generation. This research seeks to develop an accountable form of collaborative artificial intelligence within evolving cyber ecosystems.

I. INTRODUCTION

ARTIFICIAL intelligence (AI) emerges from the quest to mimic human minds through the creation of computational machinery that learns through experience and evolves beyond programmed instructions [1]. These learning machines extract meaningful representations from data, adapt their behaviours based on feedback, and ultimately develop agency for autonomous interaction with environments [2]–[10]. Yet, it remains an open question whether *general-purpose intelligence* can be realised within a solitary monolithic neural network [11]. While the *neural scaling law* suggests continuous improvements in generalisability as computational resources and corpora of knowledge increase, an ever-scaling model may still encounter inherent limitations when confronting tasks of sufficient complexity. It may struggle to decompose problems hierarchically, integrate knowledge across diverse domains and sensory modalities, sustain multiple concurrent lines of reasoning, or preserve coherence over extended inferential chains.

This calls for an account of *multi-agent collaboration*, in which each agent may possess specialised expertise,

C.-C. Chang and I. Echizen are with the Information and Society Research Division, National Institute of Informatics, Tokyo, Japan. I. Echizen is also with the Graduate School of Information Science and Technology, University of Tokyo, and the School of Multidisciplinary Sciences, Graduate University for Advanced Studies (SOKENDAI), Tokyo, Japan.

Correspondence: C.-C. Chang (email: ccchang@nii.ac.jp)

and through communication and coordination, these autonomous entities collectively tackle multifaceted tasks [12]–[18]. Against this backdrop, a fundamental question arises: as multiple agents contribute to the formation of a shared creation, how might one trace the individual contributions of each agent across the temporal dimension of generation? Without traceability, the interactions amongst multiple agents render the accountability, transparency and trustworthiness of AI fundamentally uncertain.

The concept of *provenance* resonates deeply with the fundamental human pursuit of understanding where we come from and how things came to be, addressing the primordial desire to know origins and connections across space and time. The quest of provenance is a reflection of our intrinsic curiosity about existence and continuity, manifesting across disciplines: astrophysics seeking the origins of the universe; archaeology unearthing ancient civilisations; and genetics reconstructing the evolutionary tree of life. By examining information encoded in genomes, geneticists can infer evolutionary histories, tracing the lineages of organisms across time. Likewise, when analysing a piece of writing, linguists may identify authorial traits through stylistic patterns, at times revealing signs of collaborative composition. In the realm of generative AI, however, these principles and practices face a fundamental challenge. Unlike genomic evidences left by biological evolution or linguistic clues of human authorship, the outputs of multi-agent systems may undergo complete transformation at each step. A subsequent agent may overwrite the content produced by its predecessors to maintain consistency in narrative flow, or continue the task while discarding earlier material, leaving no discernible trace of prior contributions.

One might attempt to adapt the practice of *chain of custody* from forensic science—chronological documentation recording the seizure, control, transfer and disposition of criminological evidence. In theory, such an external log could record which agent acted at each timestep, preserving a complete history of the generative process. Yet this forensic practice remains inherently fragile. External logs, or metadata, can become detached from the content it describes when transferred across platforms or corrupted during transmission, undermining the integrity of the provenance it was meant to secure. In the absence of metadata, how might provenance be preserved in a form analogous to a chain of custody, maintaining a signed and time-stamped record at each transaction amongst collaborating AI agents?

In this study, we introduce a chronological system for tracking provenance in the context of multi-agent collaboration. At its core is the concept of chronicles—symbolic

sequences that represent the chronologically ordered identities of agents throughout the generative process. The system operates through a feedback loop, in which each generative step updates the chronicle of prior interactions and synchronises it with the generated content in a steganographic manner. The scope of this study centres on generative chains of natural language created by foundation AI models. Each valid chronicle is associated with a binary codeword that defines a lexical subset of the vocabulary. The present state of the chronicle is then embedded into the generated text via a biased language generation process conditioned on the associated subset. The chronicle can subsequently be retrieved from the text through statistical analysis. This chronological system archives multi-agent provenance within the text itself, evolving alongside the act of generation without reliance on external meta-information.

II. PROVENANCE

This section reviews foundational methodologies for attributing the provenance of digital content. We examine three primary techniques, including *metadata annotation*, *fingerprinting*, and *watermarking*, outlining their principal capabilities and inherent limitations.

A. Metadata Annotation

Metadata annotation involves attaching descriptive information to data, detailing aspects such as origin of data, identity of author, time of creation, and history of usage. This archival practice facilitates data organisation and retrieval, forming the backbone of many provenance management systems [19]–[21]. Cryptographic techniques are often applied for certifying signatures [22], timestamps [23] and audit trails [24]. However, the fundamental limitation of metadata lies in its extrinsic nature. Metadata exists apart from the content it describes, making it susceptible to removal, manipulation or disassociation. This separation can compromise traceability, particularly in environments where imperfect transmission, format conversion or adversarial actions may occur.

B. Fingerprinting

Fingerprinting refers to the process of generating identifiable representations, known as fingerprints, that can uniquely identify objects [25]–[27]. Fingerprinting methods can be broadly classified into two primary categories: *cryptographic hashing* and *perceptual hashing*. Cryptographic hashing computes fixed-size digests from arbitrary-length inputs using one-way hash functions designed to be sensitive to input changes with low probability of collisions [28]–[30]. It is characterised by the avalanche effect, whereby a minimal change in the input propagates and results in a drastically different hash output. However, this sensitivity to data integrity, albeit essential for tamper-proofing, poses limitations in scenarios where content-preserving transformations are expected. Perceptual hashing extracts robust content-dependent features from data that remain stable under content-preserving transformations, thereby enabling similarity-based identification [31]–[33]. It is particularly applicable to multimedia content, which is often subject to compression, resampling, or format conversion. By determining the degree of similarity between two pieces of content, it facilitates fuzzy matching for applications such as duplicate detection and similarity search. However, this robustness to modifications comes at the cost of reduced discriminability, potentially causing collisions between different contents that share similar global structures. Cryptographic hashing offers high sensitivity for discriminability but fails under content transformations, whereas perceptual hashing provides robustness against content transformations but lacks strong guarantees of uniqueness. This trade-off between collision resistance and fault tolerance reflects a fundamental dilemma between sensitivity and robustness.

C. Watermarking

Watermarking is the practice of embedding auxiliary information into the content subject to imperceptibility constraints with respect to human perception, thereby enabling self-contained traceability without reliance on external metadata [34]–[40]. A watermark can carry provenance-related information such as unique identifiers and timestamps, providing the capability of collision resistance. In addition, it can be embedded in a way that survives common content-preserving operations such as compression, photometric distortion or geometric transformation, thus offering robust proof of provenance in multimedia distribution. The concept of watermarking has been applied to generative foundation models for provenance tracing of AI-generated synthetic content [41]. A representative methodology is based on biasing the token sampling process during text generation, controlling the distribution of selected tokens to form statistically detectable patterns [42]. It follows the principle of *zero-bit watermarking*, in which no explicit information is embedded; rather, provenance is verified by testing for the presence or absence of a given watermark. In multi-agent environments, however, zero-bit watermarking may not be directly applicable because it typically attributes each piece of content to a single source authority. While in principle it is possible for multiple watermarks to coexist within a single object without mutual interference or overwriting each other, such coexistence is not always guaranteed [43]. Even when multiple watermarks coexist, the temporal order of sequential embedding may not be reliably determined. This uncertainty calls for methodological reconsideration under circumstances where multiple agents interact and collaborate in a generative chain.

III. CHRONOLOGY

This section introduces the proposed chronological system for tracking multi-agent provenance in language generation. We formalise the notion of a chronicle, present a scalable codebook construction, describe the feedback loop for recursively updating chronicles, and detail the encoding and decoding procedures that enable post hoc recovery of generative histories from text alone.
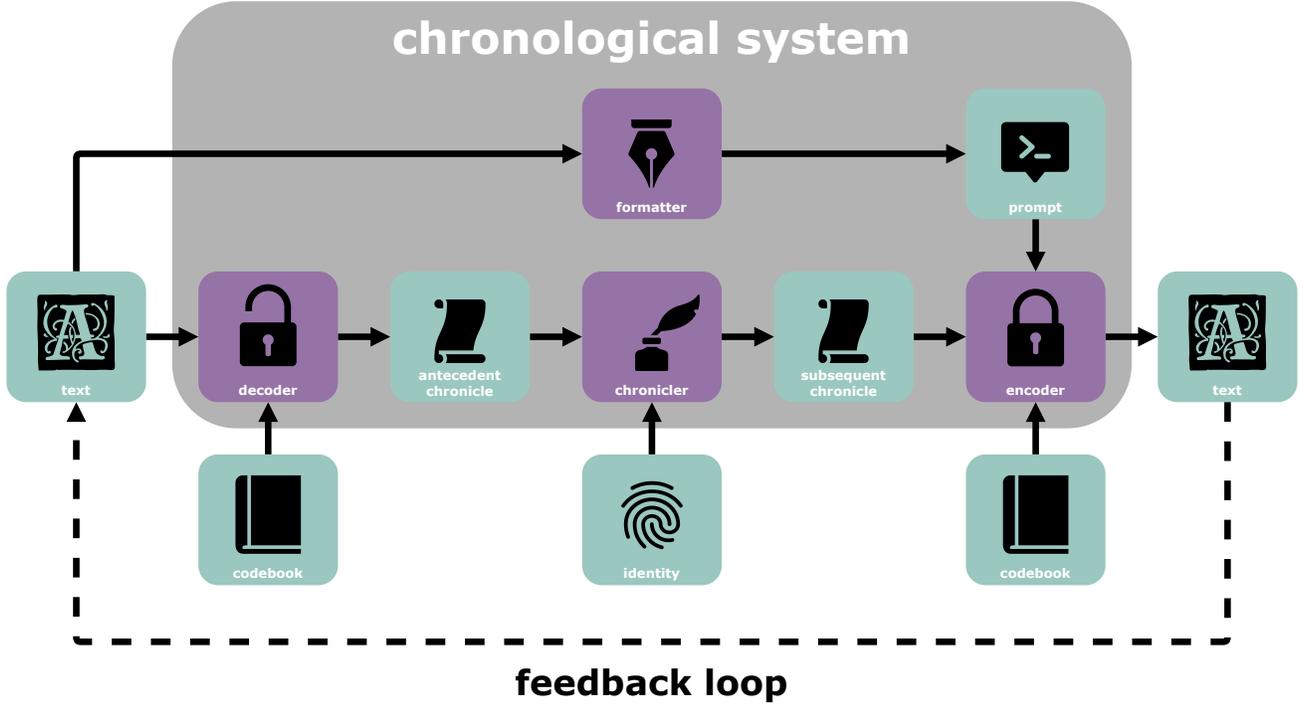
Fig. 1. Overview of the chronological system for provenance tracking through encoding, decoding and updating of chronicles within a feedback loop.

A. Chronicle Definition

Consider a set of n distinct AI agents powered by foundation models. A *chronicle* is defined as a symbolic chain of length T (i.e. the number of generative steps), represented by

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_T], \tag{1}$$

where each symbol $x_t \in \{0, 1, \ldots, n\}$ indicates the identity of the agent assigned at timestep t, and the symbol 0 denotes a null or unassigned agent. The set of all valid chronicles is defined as

$$\mathcal{X} = \{0, 1, \ldots, n\}^T. \tag{2}$$

Each unique chronicle $\boldsymbol{x}$ is associated with a corresponding binary *codeword*

$$\boldsymbol{c}(\boldsymbol{x}) \in \{0, 1\}^{|\mathcal{V}|}, \tag{3}$$

where $\mathcal{V}$ denotes the vocabulary of the underlying foundation model. The codeword marks a sparse subset of the vocabulary (e.g. with 50% of entries set to 1). The tokens corresponding to indices marked with 1 in the codeword are biased towards during generation. The collection of all codewords defines the *codebook*

$$\mathcal{C} = \{\boldsymbol{c}(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{X}\}. \tag{4}$$

The cardinality of the codebook is $|\mathcal{X}| = (n+1)^T$, encompassing all possible chronicle configurations.

B. Codebook Construction

A codebook $\mathcal{C}$ is of dimension $|\mathcal{X}| \times |\mathcal{V}|$, where $|\mathcal{X}| = (n+1)^T$ is the number of all valid chronicles and $|\mathcal{V}|$ is the vocabulary size of the foundation model. Direct generation and storage of such a codebook with unique codewords can

be computationally prohibitive and memory-intensive, particularly as the agent population, the chronicle length and the vocabulary size scale up. To keep the construction within a tractable combinatorial space, we formulate a scalable codebook generation strategy that first constructs a base codebook, where each codeword has reduced dimensionality d and then expands each codeword to match the full vocabulary size $|\mathcal{V}|$ through repetition and structured padding. In the base codebook, each codeword is generated randomly as a binary vector of constant Hamming weight $k_{\min} = \lfloor \rho \cdot d \rfloor$, where $\rho \in (0, 1)$ is a vocabulary coverage rate. A generated vector is added to the codebook only if its pattern does not duplicate any previously stored codeword, and this stochastic generation process repeats until $(n+1)^T$ unique codewords are obtained. Each base codeword is then repeated as many times as possible to reach length $|\mathcal{V}|$, with the remaining positions padded to satisfy the target Hamming weight $k = \lfloor \rho \cdot |\mathcal{V}| \rfloor$.

C. Chronological System with Feedback Loop

At each timestep t, an *antecedent chronicle* $\boldsymbol{x}^{(t-1)}$ is decoded and then updated into a *subsequent chronicle* $\boldsymbol{x}^{(t)}$ to be encoded, as illustrated in Figure 1 and presented in Algorithm 1. The process begins with an initial chronicle set to the all-zero sequence. The chronicler updates the antecedent chronicle by inserting the informed agent identity at position t, leaving the remaining entries as zeros. In other words, the subsequent chronicle is formed by concatenating a prefix of agent identities up to the preceding timestep, an infix corresponding the current agent identity, and a suffix of zero symbols to match the predefined chronicle length, yielding

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}_{1:t-1}^{(t-1)} \parallel x_t \parallel \boldsymbol{0}_{t+1:T}. \tag{5}$$
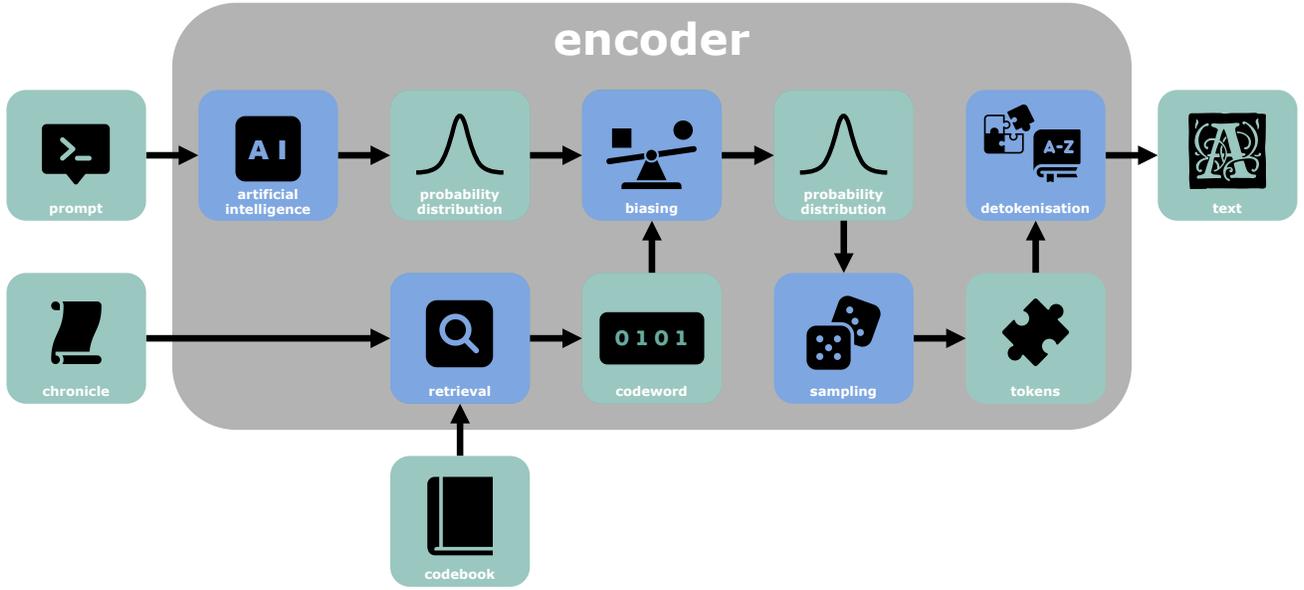
Fig. 2. Procedure of chronicle encoding, where the chronicle is embedded into the generated text through biased token sampling during language generation.
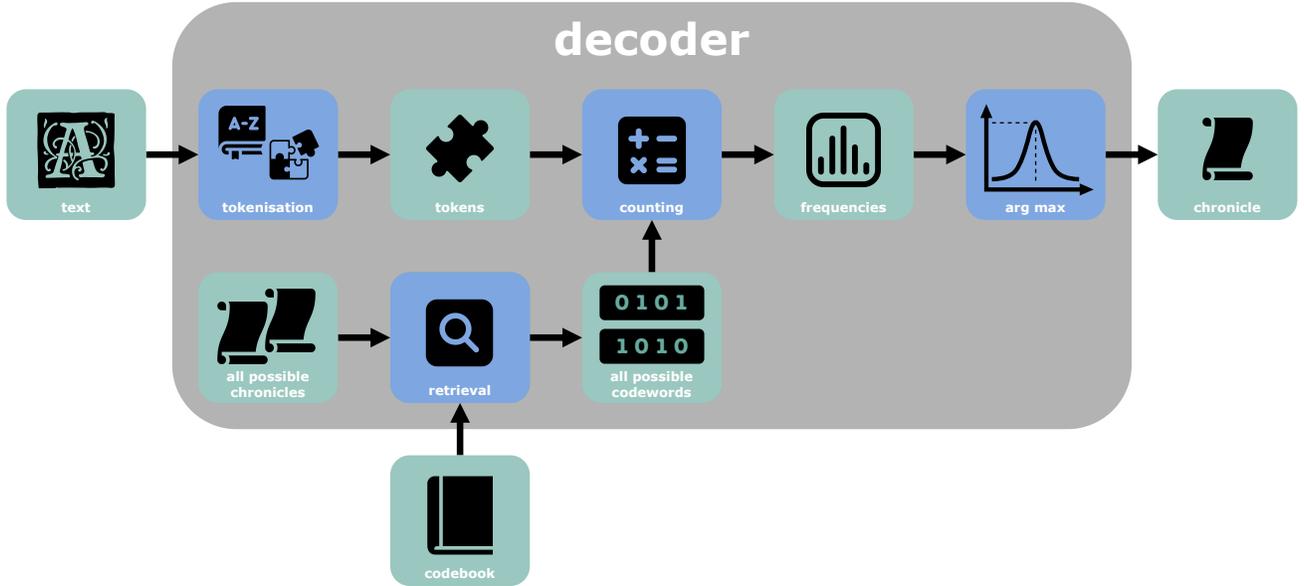

Fig. 3. Procedure of chronicle decoding, where the chronicle is retrieved from the generated text through statistical analysis of lexical choices.

The up-to-date chronicle is passed to the encoder, which embeds it into the text during the generation process of the designated agent. Once the text is generated, it is returned to the decoder, which retrieves the embedded chronicle. For the next generation step, the chronicle is updated, and the generated text is formatted into a prompt that includes the task description and, optionally, a persona that specifies the role-specific traits, behavioural constraints or stylistic preferences. The continuation of this feedback loop recursively updates the chronicle across timesteps, thereby enabling the tracking of agentic provenance throughout the generative process. Note that the update of the chronicle may optionally be performed without explicit knowledge of the current timestep; instead, the position of the next zero symbol serves as a clue for inferring

it. While the relaxation of timestep awareness offers flexibility in application scenarios where timestep tracking is unavailable or undesirable, it sacrifices fault tolerance. The update mechanism under this relaxation becomes sensitive to errors in preceding steps, where a single erroneous chronicle may trigger a cascading collapse across the remaining timesteps.

D. Chronicle Encoder

At each timestep t, the encoder receives a prompt and an up-to-date chronicle $x^{(t)}$, along with access to a predefined codebook, as illustrated in Figure 2. The codeword corresponding to the given chronicle $c(x^{(t)})$ is retrieved from the codebook. Given the prompt, the designated agent predicts an unnormalised probability distribution (i.e. logits) over the

---
Algorithm 1: Chronological System

```
// --- functions ---
Function Encoder(prompt, C, x):
    retrieve codeword c(x) from codebook C
    while terminal criterion not met do
        predict logits ℓᵥ conditioned on prompt
        apply bias according to codeword c(x):
        foreach v ∈ V do
            if cᵥ = 1 then
              | ℓ̃ᵥ ← ℓᵥ + δ;
            else
              | ℓ̃ᵥ ← ℓᵥ;
        convert logits to probabilities pᵥ = softmax(ℓ̃ᵥ)
        sample a token from probability distribution pᵥ
    detokenise tokens v to text
    return text

Function Decoder(text, C):
    tokenise text into tokens v
    generate all valid chronicles X
    foreach x ∈ X do
        retrieve codeword c(x) from codebook C
        compute frequency f(x) = Σᵥ∈ᵥ 𝕀(cᵥ = 1)
    infer likeliest chronicle x̂ = arg maxₓ f(x)
    return x̂

// --- main program ---
set vocabulary V
set number of agents n
set number of timesteps T
initialise codebook C ← {c(x) | x ∈ X}
where X = {0, 1, ..., n}ᵀ and c(x) ∈ {0, 1}^|V|
initialise chronicle x⁽⁰⁾ ← 0₁:T
for t ← 1 to T do
    assign agent identity xₜ for generation process
    construct prompt for generation process
    update chronicle:
    x⁽ᵗ⁾ ← x⁽ᵗ⁻¹⁾₁:ₜ₋₁ ‖ xₜ ‖ 0ₜ₊₁:T
    encode chronicle:
    text ← Encoder(prompt, C, x⁽ᵗ⁾)
    decode chronicle:
    x⁽ᵗ⁾ ← Decoder(text, C)
```

---

entire vocabulary. Let ℓ_v denote the logit associated with token $v \in \|\mathcal{V}\|$, and let c_v denote the corresponding binary digit in the codeword. Each logit is then selectively biased to favour tokens marked in the codeword; that is,

$$\tilde{\ell}_v = \begin{cases} \ell_v + \delta, & \text{if } c_v = 1, \\ \ell_v, & \text{otherwise,} \end{cases} \tag{6}$$

where $\delta > 0$ is a bias parameter. The biased logits are then normalised via the softmax function to obtain a probability distribution

$$p_v = \text{softmax}(\tilde{\ell}_v) = \frac{e^{\tilde{\ell}_v}}{\sum_{i=1}^{\|V\|} e^{\tilde{\ell}_i}}. \tag{7}$$

A token is sampled from the probability distribution, and the text generation process continues until an end criterion is met, such as reaching a maximum token limit or a terminal state. The resulting sequence of tokens v is then detokenised to form natural language text, within which the chronicle is implicitly embedded via lexical bias.

E. Chronicle Decoder

Given the generated text, the decoder estimates the embedded chronicle by comparing the lexical statistics computed over all possible codewords, as illustrated in Figure 3. The generated text is first tokenised into a sequence of tokens v. For each possible chronicle x, the corresponding codeword $c(x)$ is retrieved from the codebook. The frequency of matches $f(x)$ is then computed based on the number of marked tokens in the codeword that appear in the generated text; that is,

$$f(\boldsymbol{x}) = \sum_{v \in \boldsymbol{v}} \mathbb{I}(c_v = 1), \tag{8}$$

where $\mathbb{I}$ denotes the indicator function that returns 1 if the condition holds and 0 otherwise. The most probable chronicle $\hat{x}$ is then estimated by selecting the one with the maximal frequency:

$$\hat{\boldsymbol{x}} = \arg \max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}). \tag{9}$$

The decoded chronicle $\hat{x}$ serves as the reconstructed trace of agentic interactions, enabling post hoc forensic attribution from the generated text alone.

IV. EXPERIMENTS

This section evaluates the proposed chronological system through simulations of continual language generation involving multiple agents, in which the content is subject to successive transformations. We assess the complexity of the chronicle space, the accuracy of the recovered provenance, and the quality of the generated text under varying experimental conditions.

A. Experimental Setup

Data: We adopted the final paragraph from *Computing Machinery and Intelligence* by A. M. Turing as the initial seed text. At each generation step, an agent was randomly selected and prompted with the instruction: *Continue the writing from this point onwards*, as illustrated in Figure 4. This process forms a *Markovian chain* of generation, where the prompt provided to each agent depends solely on the text generated by its immediate predecessor. This experimental design reflects a continual writing process that operates without memory of the full generation history. As a result, little, if any, trace of past agent assignments is preserved in the text, posing a challenge for recovering the underlying chronology of agent participation.
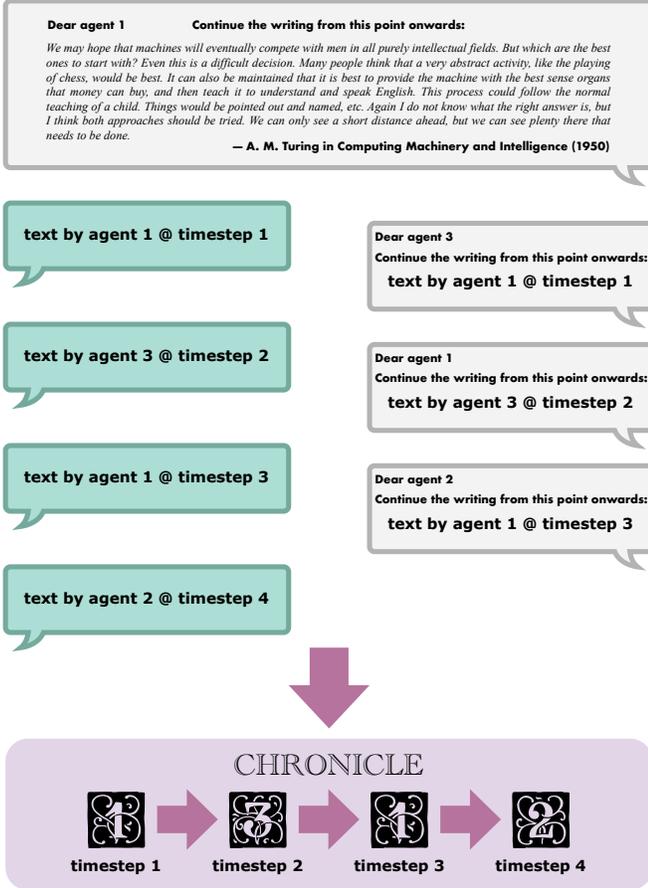
Fig. 4. Continual generative chain with multiple agents, where the chronicle is inferred post hoc from the generated content at the final timestep.

Fig. 5. Combinatorial scaling of the chronicle space $|\mathcal{X}|$ as a function of chronicle length T and agent population n.

determines the computational complexity of the decoding procedure. An exponential increase in the number of chronicles was evident as the chronicle length grows from 3 to 6 for varying numbers of agents from 2 to 4. This scaling effect reflects the combinatorial nature of multi-agent provenance tracking, where longer chronicles and larger agent populations lead to a combinatorial explosion in the size of the search space over possible chronicles.

Agents: All agents were instantiated from the same open-source foundation language model, Llama with 1 billion parameters, chosen for its efficiency and practicality in deployment on lightweight hardware. The generation process was controlled by standard sampling parameters to regulate the trade-off between diversity and coherence. The temperature was set to 0.3 to moderately sharpen the probability distribution over tokens, controlling the degree of randomness in generation. The sampling was further restricted by selecting from the top 1000 most probable tokens (hard-threshold sampling) and from the set of most probable tokens with cumulative probability more than 0.9 (nucleus sampling). The maximum token length was limited to 150 tokens per generation step.

Hyperparameters: The agent population n was varied from 2 to 4, the chronicle length T from 3 to 6, and the bias strength δ from 1 to 3. For each experimental configuration, we conducted 100 trials to obtain reliable statistics.

B. Chronicle Space Complexity

Figure 5 analyses the scaling effect of the number of valid chronicles $|\mathcal{X}|$ with respect to the chronicle length T and the agent population n. Recall that the decoding process requires an exhaustive search over the entire set of valid chronicles to infer the most probable agent sequence from the generated text. As such, the number of valid chronicles directly
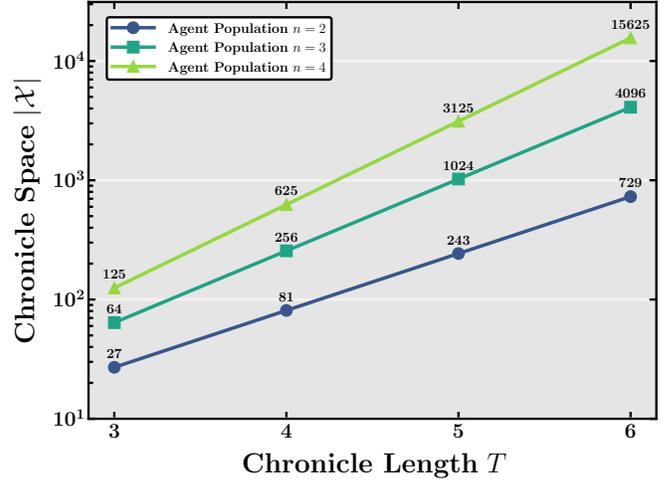
C. Chronological Accuracy

Figure 6 examines the chronological performance using a timestep-wise accuracy metric.

At each timestep t, the accuracy was computed as the proportion of correctly decoded chronicle symbols up to the current step, defined as:

$$\text{ACC} = \frac{\sum_{i=1}^{t} \mathbb{I}(\hat{x}_i = x_i)}{t}, \qquad (10)$$

where $\hat{x}_i$ and x_i denote the predicted and ground-truth symbols at timestep i, respectively, and $\mathbb{I}(\cdot)$ denotes the indicator function. This metric captures the phenomenon of error propagation, wherein an incorrectly decoded symbol is carried forward into subsequent chronicle updates and encoding steps, potentially leading to error accumulation. It was observed that the accuracy tended to decrease as the timestep increased, reflecting the challenge of preserving provenance integrity over extended generation horizons. Moreover, as the maximum chronicle length T increased, the accuracy further declined due to the exponentially expanding size of the candidate chronicle space $|\mathcal{X}|$. This degradation, nevertheless, was alleviated by increasing the bias strength δ. When the bias strength was raised from 1 to 2 or 3, near-perfect accuracy was consistently achieved across all experimental configurations, demonstrating the robustness of chronological identification under appropriate lexical bias.
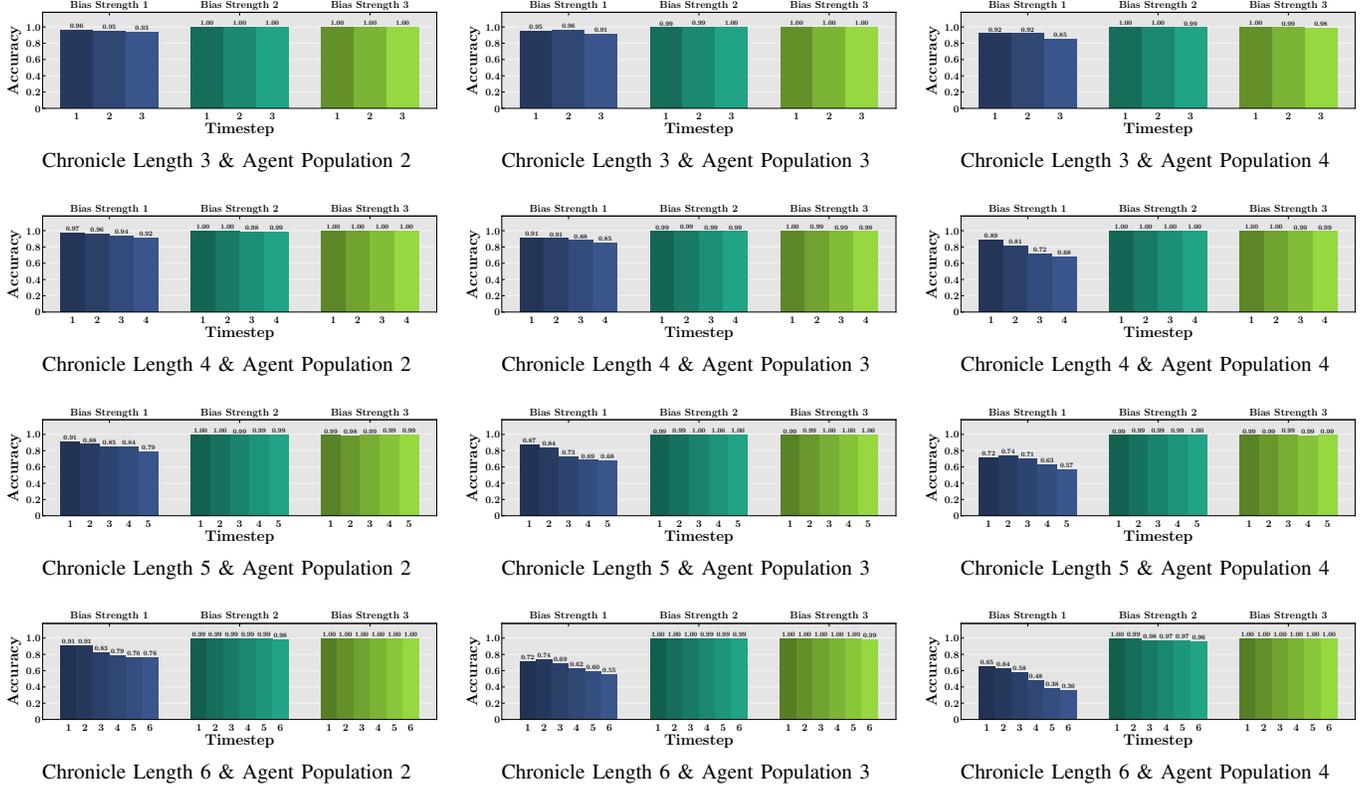
Fig. 6. Timestep-wise chronological accuracy under varying bias strengths, chronicle lengths and agent populations.
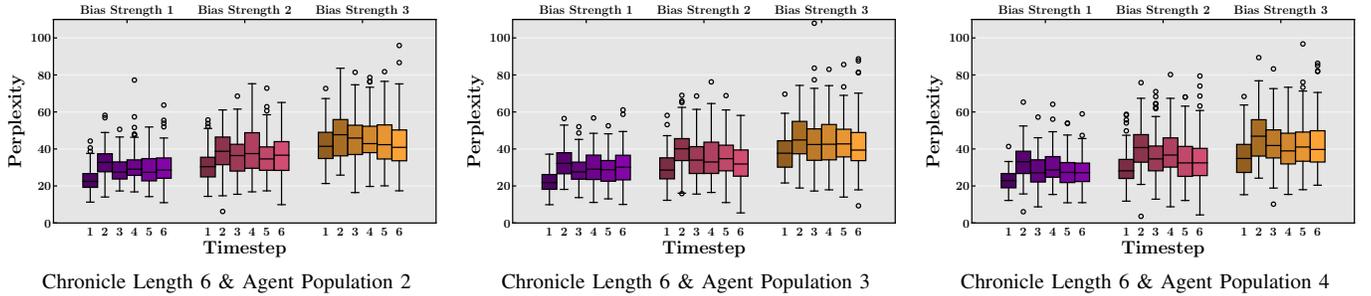

Fig. 7. Timestep-wise generative perplexity under varying bias strengths and agent populations with fixed chronicle length.

D. Generative Perplexity

Figure 7 evaluates the impact of increasing bias strength in the sampling process on generation quality. To quantify this effect, we measured the perplexity of the generated text, which captures the model's uncertainty in producing the observed sequence of tokens. Perplexity represents the inverse of the average likelihood (i.e. the geometric mean) of the predicted token probabilities over a given sequence $\boldsymbol{v}$, or equivalently, the exponentiated average negative log-likelihood, defined as:

$$\mathrm{PPL} = \left(\prod_{v \in \boldsymbol{v}} p_v \right)^{-\frac{1}{\|\boldsymbol{v}\|}} = \exp\left(-\frac{1}{\|\boldsymbol{v}\|} \sum_{v \in \boldsymbol{v}} \log p_v \right), \quad (11)$$

where p_v denotes the probability assigned by the model to token v, and $|\boldsymbol{v}|$ is the length of the sequence. Higher perplexity indicates that the model assigns lower confidence to the generated tokens, reflecting increased uncertainty or degradation in

generation quality. For this analysis, the chronicle length was fixed at 6, and perplexity was computed at each timestep under varying bias strengths and numbers of agents. As expected, increasing the bias strength led to higher perplexity, indicating a degradation in generation fluency and naturalness. While the first generation step typically exhibited lower perplexity, subsequent steps did not show a consistent increasing trend, suggesting that the impact of biased sampling stabilises after the initial generation step. Moreover, the number of agents involved did not appear to have a substantial impact on perplexity, which might imply that generation quality was primarily influenced by the bias strength rather than the complexity of the multi-agent setting.

V. Conclusion

The problem of tracking multi-agent provenance amidst the continual act of shared creation was investigated in this study. We introduced a chronological system for tracking provenance in language generation, where the history of agent contributions is not explicitly recorded as metadata but embedded within the generated content itself through the process of sampling lexical tokens. Experimental results validated the performance of the proposed system under conditions of sequential content overwriting. The combinatorial growth governed by the chronicle length and the agent population reflects the scaling complexity of the system. Furthermore, improved chronological accuracy came at the cost of increased linguistic perplexity as the bias strength was raised, revealing a trade-off between forensic traceability and generative quality. Future research may explore chronological provenance in more complex cyber ecosystems, including multimodal integration and human-in-the-loop generation, where provenance traceability constitutes a foundational tenet for trustworthy AI.

Acknowledgements

References

[1] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. 236, pp. 433–460, 1950.

[2] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities." *Proc. Natl. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, 1982.

[3] L. G. Valiant, "A theory of the learnable," *Commun. ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[4] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cogn. Sci.*, vol. 9, no. 1, pp. 147–169, 1985.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[6] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.

[7] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[8] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[9] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[10] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse control tasks through world models," *Nature*, pp. 1–19, 2025.

[11] J. A. Fodor, *The Modularity of Mind*. Cambridge, MA, USA: MIT Press, 1983.

[12] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, New Brunswick, NJ, USA, 1994, pp. 157–163.

[13] B. J. Grosz and S. Kraus, "Collaborative plans for complex group action," *Artif. Intell.*, vol. 86, no. 2, pp. 269–357, 1996.

[14] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Madison, WI, USA, 1998, pp. 746–752.

[15] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *J. Mach. Learn. Res.*, vol. 4, pp. 1039–1069, 2003.

[16] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Auton. Agents Multi-Agent Syst.*, vol. 11, no. 3, pp. 387–434, 2005.

[17] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 6382–6393.

[18] Y. Zhang, R. Sun, Y. Chen, T. Pfister, R. Zhang, and S. Arik, "Chain of agents: Large language models collaborating on long-context tasks," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 37, Vancouver, BC, Canada, 2024, pp. 132 208–132 237.

[19] G. Friedman, "The trustworthy digital camera: Restoring credibility to the photographic image," *IEEE Trans. Consum. Electron.*, vol. 39, no. 4, pp. 905–910, 1993.

[20] K.-K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. Seltzer, "Provenance-aware storage systems," in *Proc. USENIX Ann. Tech. Conf. (ATC)*, Boston, MA, USA, 2006, pp. 43–56.

[21] R. Hasan, R. Sion, and M. Winslett, "Preventing history forgery with secure provenance," *ACM Trans. Storage*, vol. 5, no. 4, pp. 1–43, 2009.

[22] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[23] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," *J. Cryptol.*, vol. 3, no. 2, pp. 99–111, 1991.

[24] B. Schneier and J. Kelsey, "Secure audit logs to support computer forensics," *ACM Trans. Inf. Syst. Secur.*, vol. 2, no. 2, pp. 159–176, 1999.

[25] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1897–1905, 1998.

[26] B. Chor, A. Fiat, M. Naor, and B. Pinkas, "Tracing traitors," *IEEE Trans. Inf. Theory*, vol. 46, no. 3, pp. 893–910, 2000.

[27] W. Trappe, M. Wu, Z. Wang, and K. R. Liu, "Anti-collusion fingerprinting for multimedia," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 1069–1087, 2003.

[28] I. B. Damgård, "A design principle for hash functions," in *Proc. Adv. Cryptol. (CRYPTO)*, Santa Barbara, CA, USA, 1989, pp. 416–427.

[29] R. C. Merkle, "One way hash functions and DES," in *Proc. Adv. Cryptol. (CRYPTO)*, Santa Barbara, CA, USA, 1989, pp. 428–446.

[30] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Proc. Adv. Cryptol. (CRYPTO)*, Santa Barbara, CA, USA, 1996, pp. 1–15.

[31] R. Venkatesan, S.-M. Koon, M. Jakubowski, and P. Moulin, "Robust image hashing," in *Proc. Int. Conf. Image Process. (ICIP)*, vol. 3, Vancouver, BC, Canada, 2000, pp. 664–666.

[32] V. Monga and B. Evans, "Perceptual image hashing via feature points: Performance evaluation and tradeoffs," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3452–3465, 2006.

[33] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *IEEE Trans. Inf. Forensics Secur.*, vol. 1, no. 2, pp. 215–230, 2006.

[34] R. van Schyndel, A. Tirkel, and C. Osborne, "A digital watermark," in *Proc. Int. Conf. Image Process. (ICIP)*, vol. 2, Austin, TX, USA, 1994, pp. 86–90.

[35] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, 1997.

[36] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proc. IEEE*, vol. 86, no. 6, pp. 1064–1087, 1998.

[37] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. IEEE*, vol. 87, no. 7, pp. 1079–1107, 1999.

[38] G. Voyatzis and I. Pitas, "The use of watermarks in the protection of digital multimedia products," *Proc. IEEE*, vol. 87, no. 7, pp. 1197–1207, 1999.

[39] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," *Proc. IEEE*, vol. 87, no. 7, pp. 1127–1141, 1999.

[40] A. Tirkel and T. Hall, "A unique watermark for every image," *IEEE Multimed.*, vol. 8, no. 4, pp. 30–37, 2001.

[41] S. Dathathri *et al.*, "Scalable watermarking for identifying large language model outputs," *Nature*, vol. 634, no. 8035, pp. 818–823, 2024.

[42] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, "A watermark for large language models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 202, Honolulu, HI, USA, 2023, pp. 17 061–17 084.

[43] A. Petrov, S. Agarwal, P. Torr, A. Bibi, and J. Collomosse, "On the coexistence and ensembling of watermarks," in *Proc. Int. Conf. Learn. Represent. (ICLR) Workshop*, Singapore, 2025, pp. 1–29.