Biweekly Report
통권 제89호(2026. 9-1)

www.copyright.or.kr

저작권 이슈 트렌드

COPYRIGHT ISSUE TREND

한국저작권위원회
KOREA COPYRIGHT COMMISSION

본 보고서는 EC21R&C(컨설팅사)에서 작성하였고, 국내외 저작권 기술·산업 동향을 조사한 자료로 한국저작권위원회 의견이 반영되어 있지 않습니다.

CONTENTS

저작권 이슈 트렌드

Biweekly Report | 통권 제89호(2026. 9-1)

생성 모델 학습을 거쳐도 살아남는 오디오 워터마크 기술

뉴스 브리프

인공지능으로 음악을 만드는 서비스가 확산되면서, 생성된 음원의 출처를 확인하려는 움직임이 나타나고 있다. 최근 한 음악 생성 기업은 자사에서 만들어진 곡에 귀로 들리지 않는 워터마크를 심고 저작권 탐지 체계를 도입하겠다고 밝혔다. 동시에 생성 모델이 특정 인물의 목소리나 기존 녹음을 동의 없이 학습하는 문제도 계속 제기되고 있다. 핵심 쟁점은 이러한 워터마크가 압축, 편집, 제거 등의 시도를 거친 뒤에도 남아 있는가, 나아가 보호된 음원이 다른 모델의 학습에 쓰였을 때 그 결과물에서도 확인이 가능한가에 있다. 본 보고서는 음성의 의미 표현에 워터마크를 심는 기술의 작동 원리를 살펴보고, 이것이 무단 학습 확인과 소유권 검증에 연결되는 방식을 분석한다.

뉴스 플러스

Ⅰ. 서론: AI 생성 음악 탐지를 위한 워터마크의 진화

• AI 생성 음악의 오용과 대응 움직임

2026년 8월, AI 음악 생성 서비스를 운영하는 수노(Suno)는 자사 플랫폼에서 만들어진 곡을 식별하기 위한 도구를 도입하겠다고 밝혔다.[1] 오디오 워터마킹(audio watermarking)과 디지털 지문 기술을 함께 활용해 다른 스트리밍 플랫폼에서 발생하는 오용을 막겠다는 구상이다. 여기에 더해 가사 데이터 기업 뮤직스매치(Musixmatch)의 저작권 탐지 시스템 센티널(Sentinel)을 도입하는 계약도 체결했다. 수노는 이러한 도구가 변조에 견디도록 설계됐으며 청취 경험을 해치지 않는다고 설명했다. 수노의 커뮤니티 가이드라인 역시 실제 인물의 목소리나 초상을 허락 없이 사용하는 행위를 금지하는 방향으로 개정됐다.

1) Ivan Mehta, “Amid legal battles, Suno says it will start watermarking songs”, TechCrunch, 2026.08.06., https://techcrunch.com/2026/08/06/amid-legal-battles-suno-says-it-will-start-watermarking-songs/

이러한 조치의 배경에는 생성된 음원이 원래 서비스를 벗어나 유통되는 과정에서 출처를 확인하기 어려워진다는 문제가 있다. 이용자가 자신이 만든 곡을 다른 플랫폼에 업로드하여 수익을 얻는 사례가 이어지면서, 어떤 음원이 어디에서 생성됐는지 가려낼 기술적 수단이 요구되고 있다. 동시에 생성 모델이 특정 창작자의 목소리나 녹음물을 동의 없이 학습에 활용했는지를 확인하려는 요구도 함께 제기된다. 결국 두 요구는 하나의 기술적 질문으로 수렴한다. 음원 안에 신호를 심어 두었을 때, 그 신호가 여러 처리와 변형을 거치고도 확인 가능한 형태로 남을 수 있는지의 문제다.

• 저수준 삽입의 한계와 의미 워터마크의 등장

오디오 워터마킹은 사람이 알아채기 어려운 신호를 음원에 삽입해 두었다가 이후 탐지기로 확인하는 기술이다. 쉽게 말하면 귀로는 구분되지 않지만 기계로는 읽어 낼 수 있는 신호를 음원 안에 넣어 두는 방식이다. 기존 방식은 이 신호를 파형이나 스펙트로그램(spectrogram)*처럼 소리를 직접 나타내는 저수준 표현(low-level representation)**에 넣어 왔다. 소리가 크고 복잡한 구간에 신호를 몰아넣으면 사람이 알아채기 어렵다는 심리음향(psychoacoustics) 원리를 활용한 결과다. 이 접근은 음질 보존에는 유리하지만, 신호가 특정 위치에 반복해서 자리 잡는다는 부작용을 낳는다.

최근 연구는 신호가 비슷한 위치에 반복된다는 점이 약점으로 작용한다고 지적한다. 공격자가 원본과 워터마크가 삽입된 음원을 함께 확보해 비교하면 신호의 형태와 삽입 위치를 학습할 수 있고, 이를 지워 내는 일이 가능해지기 때문이다. 여기에 더해 이러한 신호는 다른 모델이 해당 음원을 학습 자료로 사용했을 때 그 결과물로 이어지기 어렵다. 저수준 표현에 남은 미세한 차이는 학습 과정에서 잡음(noise)으로 처리되어 사라지는 경향이 있다. 캐나다 토론토 대학교(University of Toronto) 연구진이 제안한 람다마크(Lambda-Mark)는 이러한 한계를 넘기 위해, 저수준 표현이 아니라 음성이 담고 있는 의미 표현(semantic representation)에 워터마크를 심는 방향을 택했다.

* 스펙트로그램(spectrogram): 소리를 여러 주파수 성분으로 나눈 뒤, 각 성분의 세기가 시간에 따라 어떻게 변하는지를 시각화한 자료
** 저수준 표현(low-level representation): 소리의 의미나 내용을 따로 정리하지 않고, 파형이나 스펙트로그램처럼 물리적 신호에 가까운 형태로 나타낸 것

II. 본론: 의미 표현을 활용한 오디오 워터마킹 기술

• 오디오 워터마크의 견고성과 방사성 요건

워터마크가 실제로 쓸모를 가지려면 두 가지 성질을 함께 갖춰야 한다. 첫째는 견고성(robustness)으로, 음원이 압축이나 잡음 추가, 재샘플링 같은 처리를 거치거나 누군가 의도적으로 신호를 지우려 시도한 뒤에도 여전히 탐지되는 성질을 뜻한다. 쉽게 말하면 음원이 이런저런 손질을 거쳐도 심어 둔 신호가 살아남아 읽히는 상태다. 둘째는 방사성(radioactivity)으로, 워터마크가 삽입된 음원으로 다른 생성 모델을 학습시켰을 때 그 모델이 만들어 낸 결과물에서도 신호가 확인되는 성질이다. 이는 신호가 음원 한 개에 머무르지 않고, 그 음원을 학습한 모델의 산출물로까지 옮겨 간다는 의미다.

두 성질이 함께 요구되는 이유는 위협 발생 지점이 서로 다르기 때문이다. 견고성만 갖춘 워터마크는 음원 자체가 유통될 때 출처를 밝히는 데는 쓰이지만, 그 음원이 학습에 쓰인 뒤 만들어진 산출물과의 연결을 보여주지는 못한다. 반대로 방사성만 갖춘 워터마크는 학습을 거쳐 워터마크가 전달되더라도, 산출물에 간단한 변형이 가해지면 워터마크가 쉽게 훼손되어 식별이 어려워진다. 성우의 목소리가 무단 복제되거나 녹음이 동의 없이 학습에 사용되는 상황이라면 두 지점 모두가 방어 대상에 들어온다. 또한, 여러 정보를 담을 수 있는 워터마크는 누구의 음원인지를 표시할 수 있어, 어느 경로로 자료가 유출됐는지 찾는 데 활용될 수 있다.

기존 워터마킹 방식이 이 요건을 동시에 충족하지 못한 배경에는 음질을 지키려는 설계 선택이 자리한다. 신호를 사람이 인지하기 어려운 시간-주파수 구간(time-frequency bin)에 집중해서 넣다 보니, 삽입 위치와 형태가 음원마다 비슷하게 반복되는 결과로 이어졌다. 이렇게 되면 여러 음원을 모아 비교하는 방식으로 신호의 규칙을 파악할 수 있고, 파악된 규칙은 곧 제거의 실마리가 된다. 학습 과정에서도 마찬가지 문제가 나타난다. 생성 모델은 저수준 표현보다 음성의 전반적인 특징을 익히는 방향으로 훈련되기 때문에, 저수준 표현에 얹힌 미세한 차이는 학습에 반영되지 않고 걸러지기 쉽다.

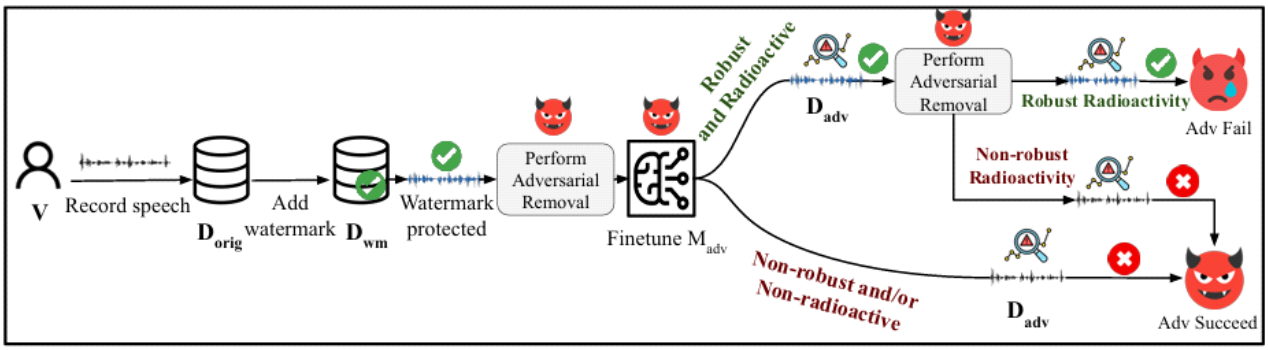
• 의미 잠재 표현에 신호를 심는 삽입 방식

람다마크는 파형이 아닌 의미 잠재 표현(semantic latent representation)에 워터마크를 삽입한다. 의미 잠재 표현은 미리 학습된 오디오 인코더(audio encoder)가 음원을 입력받아 만들어 내는 숫자 형태의 표현으로, 발음 내용이나 억양, 음색처럼 소리가 담고 있는 고수준 속성을 압축해 담고 있다. 쉽게 말하면 파형이 소리의 물리적 파동을 그대로 적어 둔 기록이라면, 의미 잠재 표현은 그 소리가 무엇을 말하고 어떤 목소리인지를 요약해 둔 기록이다. 워터마크를 이 층위에 심는다는 것은 저수준 표현을 미세하게 건드리는 대신, 요약된 내용 자체를 일정한 방향으로 조금씩 밀어 놓는다는 뜻이다.

삽입 과정은 단계를 따라 진행된다. 먼저 워터마크 인코더가 담고자 하는 메시지를 입력받아 잠재 공간 안에서 하나의 방향을 만들어 낸다. 이 방향은 원본 음원의 내용과 무관하게 메시지에 의해서만 결정되며, 모든 프레임에 동일하게 더해진다. 그 결과 서로 다른 음원에 워터마크를 넣더라도 동일한 방향의 변화가 일관되게 나타나고, 이는 뒤이어 살펴볼 방사성의 토대가 된다. 이렇게 변형된 잠재 표현을 보코더(vocoder)가 다시 음원으로 되돌리면 워터마크가 담긴 음원이 완성된다. 탐지 단계에서는 이 음원을 다시 인코더에 통과시켜 잠재 표현을 얻은 뒤, 디코더가 워터마크의 존재 여부와 담긴 메시지를 함께 읽어 낸다.

신호의 크기를 정하는 방식에도 조정이 들어간다. 일정한 크기로만 신호를 더하면 소리가 크고 복잡한 음원에서는 신호가 묻혀 읽히지 않고, 조용한 음원에서는 지나치게 두드러져 음질을 해친다. 람다마크는 원본 잠재 표현의 크기에 비례해 신호의 세기를 조절하며, 원본 음원에 적합한 신호 크기의 비율은 학습 과정에서 스스로 찾아낸다. 다만 비율에는 상한을 두어 탐지 성능을 높이려고 신호를 무한정 키우지 않도록 제한한다. 이러한 설계는 음질 보존과 탐지 성능이라는 서로 상충되는 조건 사이에서 균형점을 잡기 위한 장치로 볼 수 있다.

[그림 1] 워터마크의 견고성과 방사성에 따른 확인 결과의 차이



출처: Kexin Li 외 3인, "LambdaMark: Semantic Audio Watermarking for Robustness and Radioactivity", arXiv, 2026.06.19., https://arxiv.org/pdf/2606.21365

• **다운스트림 학습을 견디는 방사성 신호**

의미 잠재 표현에 삽입된 워터마크는 학습을 거쳐 다른 모델의 산출물로 옮겨 갈 수 있다. 앞서 살펴본 것처럼 람다마크는 모든 음원에 동일한 방향의 변화를 일관되게 더한다. 보호된 음원 여러 개를 모아 생성 모델을 추가 학습시키면, 모델은 이 변화를 개별 음원의 잡음이 아니라 학습 자료 전반에 반복되는 특징으로 받아들이게 된다. 이러한 점에서 워터마크는 걸러 낼 잡음이 아니라 익혀야 할 성질의 위치에 놓인다. 그 결과 학습을 마친 모델이 만들어 낸 산출물에서도 워터마크가 다시 확인된다.

이 성질은 자신의 음원이 무단으로 학습되었는지 확인하려는 창작자의 권리 보호와 연결되는 지점에서 차이를 만든다. 상용 생성 모델은 어떤 자료로 학습했는지를 공개하지 않는 경우가 많아, 특정 음원이 학습에 쓰였는지 외부에서 확인하기 어렵다. 방사성을 갖춘 워터마크는 모델의 내부 구조나 학습 방식에 접근하지 않고도, 그 모델이 만들어 낸 음원만 확보해 신호의 존재를 확인하는 방식으로 이 문제에 접근한다. 창작자 입장에서는 자신이 보호 조치를 적용해 둔 음원과 문제가 된 산출물 사이의 연관성을 기술적으로 제시할 수 있는 근거가 확보되는 셈이다. 다만 이러한 확인 결과가 곧바로 권리 침해에 대한 판단으로 이어지는 것은 아니며, 어디까지나 사실관계를 좁히는 단서로 기능한다.

여러 비트의 정보를 담을 수 있다는 특성은 여기서 또 다른 활용으로 이어진다. 워터마크에 소유자를 구분하는 정보를 함께 넣어 두면, 산출물에서 읽어 낸 값을 통해 보호되는 음원 가운데 어느 것이 유출됐는지 특정할 수 있다. 이는 여러 창작자가 각기 다른 음원을 보호하고 있는 상황에서 유출 경로를 추적하는 데 활용될 가능성이 있다. 워터마크가 파형의 특정 위치에 얽혀 있지 않고 음원 전체에 퍼진 형태로 남는다는 점은 산출물 단계에서도 이어진다. 지워야 할 대상이 한곳에 모여 있지 않으므로, 산출물에 변형을 가해 워터마크를 지우려는 시도 역시 같은 이유로 어려워진다.

• 다양한 공격과 생성 모델에서의 검증 결과

람다마크의 성능은 음원에 변형이 가해진 상황과 신호를 의도적으로 지우려는 시도가 있는 상황으로 나누어 확인됐다. 잡음 추가나 재샘플링, 음량 조절, 필터링, 위상 변경처럼 실제 유통 과정에서 흔히 발생하는 처리가 적용됐고, 압축 코덱을 거치게 하거나 학습을 활용해 신호를 지우려는 시도도 함께 적용됐다. 기존 방식들은 특정 처리에는 잘 버텼지만 다른 처리에서는 탐지가 사실상 무작위 추측 수준으로 떨어지는 모습을 보였다. 특히 위상을 흔들거나 잡음을 더하는 비교적 단순한 처리에서도 무너지는 사례가 확인됐다. 반면 람다마크는 평가한 모든 처리와 제거 시도에서 탐지가 유지된 유일한 방식으로 나타났으며, 학습에 쓰이지 않은 다른 음성 자료에 적용했을 때도 비슷한 수준을 보였다.

음질 측면에서는 평가 방식에 따라 결과가 갈렸다. 원본과 직접 비교하는 지표에서는 기존 방식보다 낮은 값이 나왔는데, 이는 람다마크가 파형을 그대로 두는 대신 의미 표현을 조금씩 밀어 놓기 때문에 예상되는 결과다. 반대로 원본 없이 소리 자체의 자연스러움을 평가하는 지표에서는 기존 방식보다 높은 값을 기록했고, 음성인식 모델로 내용을 옮겨 적어 원문과 비교했을 때도 발화 내용이 거의 그대로 보존됐다. 방사성 검증에서는 성격이 다른 세 종류의 생성 모델이 대상이 됐고, 모델 전체와 일부만 조정하는 두 방식 모두에서 산출물에서 신호가 확인됐다. 학습 자료의 규모를 줄여도 신호가 전달되는 경향이 유지됐으며, 산출물에 제거 시도를 가한 뒤에도 탐지가 이어졌다.

III. 결론: 워터마크 삽입 층위의 변화와 검증 과제

• 삽입 층위 전환과 확인 절차의 변화

결국 람다마크는 워터마크를 심는 층위를 바꾸면 기술의 성질도 달라진다는 점을 보여 준다. 신호를 파형의 특정 구간에 몰아넣지 않고 의미 표현에 얹으면, 지워야 할 대상이 흩어져 제거가 어렵고 학습 과정에서도 걸러지지 않는다. 이를 통해 확인 범위는 음원 자체에서 그 음원을 학습한 모델의 산출물로 넓어진다. 이러한 점에서 학습 자료가 공개되지 않는 조건에서도 관계 확인이 가능해진다. 이는 산출물을 놓고 유사성을 따지는 대신, 원본에 워터마크를 미리 심어 두고 나중에 판독하는 방식임을 의미한다.

따라서 남은 과제는 이 방식이 어느 범위까지 적용되는지를 확인하는 일에 놓인다. 음성을 이산 토큰(discrete token)으로 바꾸어 다루는 생성 모델에서는 신호가 부분적으로만 전달되는 것으로 나타났으며, 신호의 세기를 키워 대응할 경우 음질이 함께 떨어지는 관계가 확인됐다. 이는 모델의 처리 방식에 맞추어 삽입 설계를 조정할 필요가 있음을 뜻한다. 또한 이 방식은 창작자가 원본에 미리 워터마크를 심어 두고, 그 자료가 학습에 사용됐는지를 이후에 확인하는 데 초점을 둔다. 이러한 접근이 실제로 자리 잡으려면 원본을 보호하는 단계와 산출물을 확인하는 단계가 함께 이어질 필요가 있다.

* 이산 토큰(discrete token): 이어지는 소리를 잘게 나눈 뒤, 정해진 종류의 조각 가운데 하나로 바꾸어 나타낸 단위

참고문헌

• Kexin Li 외 3인, "LambdaMark: Semantic Audio Watermarking for Robustness and Radioactivity", arXiv, 2026.07.19., https://arxiv.org/pdf/2606.21365

기술용어

순번	용어	설명
1	**스펙트로그램** **(spectrogram)**	소리를 여러 주파수 성분으로 나눈 뒤, 각 성분의 세기가 시간에 따라 어떻게 변하는지를 시각화한 자료
2	**저수준 표현** **(low-level representation)**	소리의 의미나 내용을 따로 정리하지 않고, 파형이나 스펙트로그램처럼 물리적 신호에 가까운 형태로 나타낸 것
3	**이산 토큰** **(discrete token)**	이어지는 소리를 잘게 나눈 뒤, 정해진 종류의 조각 가운데 하나로 바꾸어 나타낸 단위

두 신호로 심는 상보 워터마크, 출처와 변조 가르는 3단계 판정 설계

뉴스 브리프

AI가 만든 텍스트에 사람이 알아볼 수 없는 신호를 심어 그 출처를 확인하는 워터마크 기술이 산업 전반으로 확산되고 있다. 앤트로픽은 EU의 인공지능법에 대응하기 위해 자사 모델이 처리한 텍스트 전반에 워터마크를 적용한다고 밝혔으며, 현재 그 적용 범위 및 해석을 둘러싼 논의가 이어지고 있다. 핵심 쟁점은 워터마크가 편집을 견디도록 설계될수록 그 텍스트가 중간에 바뀌었는지를 가려내기 어려워진다는 점에 있다. 편집에 강한 신호는 내용이 일부 바뀐 뒤에도 출처를 그대로 인정하기 때문이다. 본 보고서는 강건 신호와 취약 신호를 한 텍스트에 함께 심어 출처 추적과 변조 탐지를 동시에 수행하는 워터마크 기법의 작동 원리를 살펴보고, 앞으로 필요한 대응 방향을 전망한다.

뉴스 플러스

Ⅰ. 서론: AI 생성 텍스트 식별 기술의 확산

- **클로드 워터마크 도입과 워터마크 범위 논란**

미국 생성형 AI 서비스 개발기업 앤트로픽(Anthropic)은 자사 언어모델 클로드(Claude)가 산출하는 텍스트에 육안으로는 확인할 수 없는 워터마크를 삽입한다고 밝혔다.[2] 이 조치는 유럽연합(European Union, 이하 EU) 인공지능법(AI Act)이 요구하는 AI 생성물 표시 의무에 대응하기 위한 것으로, EU 역내에 한정되지 않고 전 세계에 제공되는 신규 모델 전반에 처음부터 적용된다. 적용 대상에는 대화형 서비스 클로드뿐만 아니라 코딩 도구인 클로드 코드(Claude Code), 개발자용 인터페이스인 클로드 플랫폼 API(Claude Platform API)를 통해 생성된 텍스트가 함께 포함되며, 이미지 파일에는 텍스트와 별개로

2) Ashley Belanger, "Claude's new Scarlet Letter watermark is invisible—for now", Ars Technica, 2026.08.13., https://arstechnica.com/tech-policy/2026/08/claudes-new-scarlet-letter-watermark-is-invisible-for-now/

콘텐츠(content)의 생성과 편집 이력을 기록하는 개방형 규격인 C2PA(Coalition for Content Provenance and Authenticity)* 메타데이터가 첨부된다. 이 워터마크는 문장의 의미, 품질, 가독성에는 영향을 주지 않으면서 기계만이 읽어 낼 수 있는 신호로 텍스트 안에 남는다.

논의의 초점은 워터마크가 붙는 대상이 모델이 새로 만들어 낸 문장에 국한되지 않고, 사람이 작성한 글을 교정하거나 번역하거나 요약한 결과물에까지 남는다는 데 있다. 이 경우 워터마크는 클로드가 해당 텍스트를 처리했다는 사실을 알려줄 뿐 내용을 누가 썼는지는 구분하지 못한다. 앤트로픽은 워터마크가 작성 주체가 아니라 처리 이력을 나타낸다고 설명하고 있으며, 탐지 방법과 오탐 가능성에 관한 검증 자료를 공개하지 않아, 워터마크의 신뢰도를 외부에서 확인하기 어렵다는 문제가 남는다.

* C2PA(Coalition for Content Provenance and Authenticity): 콘텐츠의 출처와 제작 이력을 확인할 수 있는 공통 규격을 개발하기 위해 여러 기업과 기관이 결성한 연합체

• 출처 추적과 변조 탐지가 부딪히는 지점

워터마크는 언어모델이 다음 단어를 고를 때 참조하는 확률값을 비밀 키(secret key)에서 만들어 낸 신호에 맞추어 조정하는 방식으로 작동한다. 이 신호는 어휘 전체를 두 묶음으로 나누고 한쪽 묶음에 속한 단어가 더 자주 선택되도록 확률을 끌어올리며, 텍스트가 그 묶음을 얼마나 많이 따랐는지를 수치로 계산한다. 계산된 수치가 미리 정한 기준값을 넘으면 해당 텍스트는 워터마크를 적용한 모델에서 나온 것으로 판정되며, 이것이 출처 추적의 기본 절차다.

워터마크에 가해지는 첫 번째 위협은 제거(removal)인데, 이는 공격자가 문장의 일부를 바꾸어 탐지를 피하려는 시도를 말한다. 이에 대응해 기존 기법들은 편집을 겪어도 신호가 유지되도록 강건성을 높이는 방향으로 발전해 왔다.

그러나 신호가 편집을 잘 견딜수록 내용이 바뀐 텍스트에도 출처 식별이 그대로 유지되어 새로운 공격점이 될 수 있다. 워터마크가 적용된 문장에서 몇 개의 단어만 바꾸어 문장의 의미를 뒤집어도 신호는 여전히 강하게 남으며, 탐지기는 모델이 만들지 않은 내용을 모델의 산출물로 인정하게 된다. 이러한 공격을 편승 위조(piggyback spoofing)라 하며, 이를 막으려면 편집이 가해졌을 때 곧바로 신호가 깨져야 한다. 출처 추적은 편집을 견디는 신호를 요구하고, 변조 탐지는 편집에 깨지는 신호를 요구하는데, 지금까지의 기법들은 하나의 텍스트에 신호를 하나만 심어 왔기 때문에 두 요구를 동시에 충족하기 어려운 상태다.

II. 본론: 상호 보완 신호의 작동 원리

• 강건 신호와 취약 신호를 나누는 윈도 길이

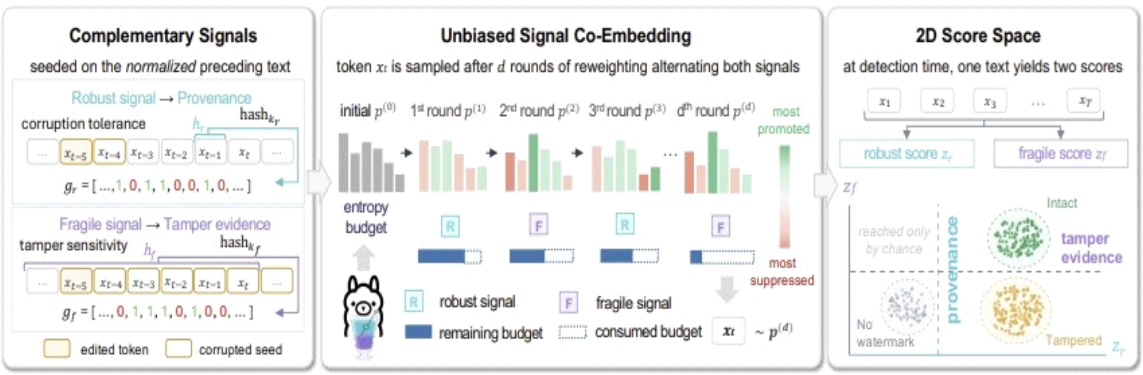
상보 워터마크(complementary watermark) 기법은 하나의 텍스트에 성격이 다른 두 신호를 함께 심는다. 하나는 편집을 견디도록 설계된 강건 신호(robust signal)로 출처 추적을 맡고, 다른 하나는 편집에

쉽게 깨지는 취약 신호(fragile signal)로 변조 탐지를 맡는다. 두 신호는 어휘 전체를 두 묶음으로 나누어 한쪽 단어의 선택 확률을 올리고 다른 쪽을 낮추는 동일한 방식으로 만들어지며, 서로 다른 비밀 키를 사용하기 때문에 통계적으로 독립된 신호가 된다. 두 신호를 가르는 것은 신호를 계산할 때 참조하는 앞부분 텍스트의 범위, 즉 윈도(window)의 길이 하나뿐이다.

모델은 단어 하나를 고를 때마다 바로 앞에 놓인 텍스트를 해시 함수에 넣어 그 자리에서 쓸 신호를 만들어 낸다. 해시 함수에 들어가는 입력값을 시드(seed)라 하며, 텍스트가 바뀌면 시드도 함께 바뀌어 그 자리의 신호가 처음 생성 시점과 달라진다. 어떤 위치에 편집이 가해지면 그 지점부터 윈도 길이만큼 시드 불일치가 발생한다. 이때 윈도가 짧으면, 불일치가 소수 단어에 국한되지만, 윈도가 길면 그 영향이 후속 텍스트까지 확산된다.

기존 워터마크 기법들은 윈도 길이를 대체로 단어 한 개에서 세 개 사이로 설정해 왔으며, 이는 편집에 강한 쪽에만 치우쳐 있었음을 의미한다. 편집을 견디는 성질은 문장을 수정하거나 번역을 거쳐 유통되는 실제 환경에서 출처를 식별하기 위해 요구되는 조건이었다. 다만 이러한 특성으로 인해 내용이 바뀐 텍스트에도 출처가 그대로 인정되며, 워터마크는 모델이 만들지 않은 내용도 보증하게 된다.

[그림 1] 상호 보완 신호가 작동하는 원리



출처: Xiaoyan Feng 외 4인, "Tracing Provenance and Detecting Tampering with Complementary LLM Watermarks", arXiv, 2026.08.13., https://arxiv.org/pdf/2608.12713

반면 상보 워터마크 기법은 강건 신호에 바로 앞 단어 하나만 보는 짧은 윈도를 주고, 취약 신호에는 그때까지 생성된 앞부분 전체를 문자 단위로 훑는 긴 윈도를 준다. 이처럼 윈도 길이를 두 갈래로 나누어 배치하면 출처 확인 기능을 유지한 채 텍스트가 전달 과정에서 손상되었는지를 별도로 읽어 낼 수 있고, 워터마크의 유무만 가리던 탐지가 텍스트의 상태까지 가리는 방식으로 바뀌는 것이다.

• **정규화된 텍스트를 기준으로 삼는 신호 설계**

기존 워터마크 기법 대부분은 모델 내부에서 쓰는 토큰 ID(token ID)를 시드의 재료로 삼는다. 토큰(token)은 모델이 텍스트를 다룰 때 쓰는 최소 단위로, 단어 하나가 통째로 한 토큰이 되기도 하고 긴 단어가 여러 조각으로 나뉘기도 한다. 상보 워터마크 기법은 토큰이 신호를 실어 나르는 역할을 그대로 하도록 두되, 정규화된 텍스트(normalized text)를 시드로 사용한다. 토큰 ID는 생성 과정에서 잠시 쓰이는 중간 매개체일 뿐이고, 독자가 받아 보는 것은 텍스트 자체이기 때문이다.

정규화(normalization)는 겉보기에 같은 글자를 하나의 표준 형태로 모으는 처리를 뜻하며, 화면에 드러나지 않는 차이를 지우는 과정으로 이해할 수 있다. 정규화는 폭 없는 공백(zero width)*과 보이지 않는 부호를 제거하고, 유니코드(Unicode)** 호환 형식을 표준 형식으로 바꾸며, 동형 문자 (homoglyph)***를 표준 문자로 통합하고, 대소문자를 통일한 뒤, 여러 개로 이어진 공백을 하나로 줄이는 순서로 작동한다.

정규화를 신호를 만드는 단계에 두는지 탐지 단계에만 두는지의 여부에 따라 결과가 달라진다. 실험에서 정규화를 전혀 적용하지 않은 경우, 동형 문자로 30%를 바꾼 텍스트의 출처 식별 성공률은 33.6%까지 떨어졌고, 정규화를 탐지 단계에만 적용한 경우에는 출처 식별은 회복되었으나 변조 탐지율이 43.5%로 감소했다.[3] 반면 신호 생성 단계부터 정규화를 적용하면 출처 식별은 100%를 유지하고 변조 탐지율도 98.3%에 이른다. 이는 워터마크가 무엇을 보증하는지를 결정하는 것과 연결된다. 눈에 보이지 않는 문자를 교체하는 것만으로 출처 식별이 무너진다면 어디까지가 원래 산출물이고 어디부터가 변형인지를 기술적으로 가려내기 어려워진다.

* 폭 없는 공백(zero width): 화면에 글자나 공간을 차지하지 않으면서 단어의 경계나 줄 바꿈 위치를 지정하는 데 쓰이는 보이지 않는 특수 문자
** 유니코드(Unicode): 전 세계의 모든 문자를 컴퓨터에서 똑같이 표현하기 위해 만든 국제 표준 문자 체계
*** 동형 문자(homoglyph): 모양이 아주 비슷하거나 똑같아서 눈으로 구별하기 어려운 글자. 대표적인 사례는 숫자 0과 대문자 O, 영어 알파벳 a와 키릴문자 а

• **생성 품질을 유지하는 무편향 재가중 방식**

두 신호가 각각 제 역할을 하려면, 생성되는 모든 토큰에 두 신호가 함께 실려야 한다. 한 토큰에 신호 하나만 배정하는 방식도 생각할 수 있으나, 이 경우 탐지기는 어떤 토큰이 어느 신호를 담고 있는지 알 수 없어, 각 신호의 입장에서는 절반의 자리가 다른 신호의 잡음(noise)으로만 남는다. 실험에서 한 토큰에 신호 하나만 배정한 경우, 원본 그대로인 텍스트의 강건 점수가 9.4에서 6.1로 낮아졌고, 토큰 50개 길이에서 출처 식별 성공률이 98.3%에서 86.0%로 내려갔다. 두 신호를 한 토큰에 겹쳐 심는 방식이 두 목표를 함께 달성하는 데 필요한 조건임을 의미한다.[4]

3) Xiaoyan Feng 외 4인, "Tracing Provenance and Detecting Tampering with Complementary LLM Watermarks", arXiv, 2026.08.13., https://arxiv.org/pdf/2608.12713
4) Xiaoyan Feng 외 4인, "Tracing Provenance and Detecting Tampering with Complementary LLM Watermarks", arXiv, 2026.08.13., https://arxiv.org/pdf/2608.12713

문제는 같은 토큰의 확률 분포를 두 번 조정하면 생성 품질이 흔들린다는 점에 있다. 상보 워터마크 기법은 무편향 재가중(unbiased reweighting)을 사용해 이 문제를 다룬다. 무편향 재가중은 확률 분포를 조정하되 조정 결과의 기댓값이 원래 분포와 같아지도록 하는 방식이며, 개별 텍스트에서는 단어 선택이 신호 쪽으로 기울지만, 여러 번 생성한 결과를 평균 내면 원래 모델의 분포와 일치한다는 뜻이다. 구체적으로는 한 라운드마다 어휘를 절반씩 나누는 묶음을 새로 뽑아 한쪽의 확률을 올리고 다른 쪽을 낮추는데, 올리는 몫과 낮추는 몫이 서로 상쇄되도록 계산해 기댓값이 유지된다. 라운드마다 뽑는 묶음이 서로 독립이므로 여러 라운드를 거쳐도 이 성질은 그대로 남는다.

라운드를 두 신호에 어떻게 나누어 주느냐가 두 목표 사이의 균형을 조절하는 장치가 된다. 실험에서 토큰 하나당 30라운드를 적용하고 이를 1대 1, 2대 1, 4대 1의 비율로 강건 신호와 취약 신호에 배분한다. 각 라운드는 원래 분포가 지닌 여유분을 나누어 쓰기 때문에 배분 비율이 대체로 신호 세기의 비율로 이어지며, 강건 신호 쪽 라운드를 늘리자, 출처 식별은 82.0%에서 92.4%로 올라가고 변조 탐지는 99.4%에서 97.2%로 내려간다. 이처럼 서비스 제공자는 유통 환경에 맞추어 출처 확인과 변조 확인 가운데 어느 쪽에 비중을 둘지 조절할 수 있다.[5]

• 두 점수로 결정되는 3단계 판정과 그 조건

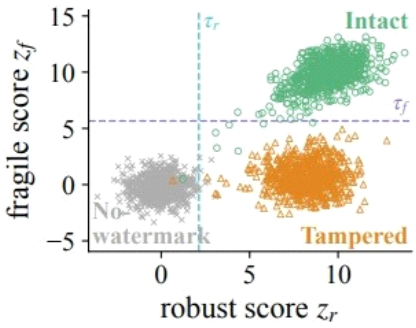
탐지 단계에서는 텍스트를 다시 토큰으로 나눈 뒤, 실제로 쓰인 단어가 선택 확률이 높아진 묶음에 몇 번이나 들어갔는지를 신호별로 센다. 워터마크가 없는 텍스트라면 각 자리의 적중 여부가 절반의 확률로 독립적으로 정해지므로, 관측된 횟수가 절반에서 얼마나 벗어났는지를 표준화한 값이 그대로 신호의 세기가 된다. 이렇게 강건 점수와 취약 점수 두 개가 나오고, 두 점수는 하나의 평면 위에 텍스트의 위치를 찍는 좌표로 쓰인다. 강건 점수는 출처를, 취약 점수는 내용의 동일성을 의미한다.

3단계 판정(three-state decision)은 이 평면을 세 구역으로 나누어 읽는 방식이다. 취약 점수가 낮은 텍스트는 워터마크가 없는 텍스트일 수도 있고 워터마크가 있었으나 편집으로 시드가 어긋난 텍스트일 수도 있어 그 자체로는 구분되지 않으며, 강건 축(robust axis)으로 모호함을 판단한다.[6] 강건 점수가 기준값을 넘지 못하면 '워터마크 없음(No-Watermark)'으로 판정하고, 기준값을 넘긴 경우에만 취약 점수를 읽어 기준값을 넘으면 '원본 그대로임(Intact)', 넘지 못하면 '변조됨(Tampered)'으로 판정한다.

이 방식이 성립하려면 두 신호가 편집에 서로 다르게 반응해야 한다. 취약 신호의 윈도를 강건 신호와 같은 길이로 줄이면 두 점수가 같은 정보를 담게 되어 변조된 텍스트와 원본을 가르는 경계가 사라진다. 실험에서 취약 윈도를 문자 40개로 두었을 때 변조 탐지율은 7.0%에 그쳤으나, 200개로 늘리자 88.0%까지 올라갔다.

5) Xiaoyan Feng 외 4인, "Tracing Provenance and Detecting Tampering with Complementary LLM Watermarks", arXiv, 2026.08.13., https://arxiv.org/pdf/2608.12713
6) Xiaoyan Feng 외 4인, "Tracing Provenance and Detecting Tampering with Complementary LLM Watermarks", arXiv, 2026.08.13., https://arxiv.org/pdf/2608.12713

[그림 2] 3단계 판정에서 강건 점수와 취약 점수에 따른 텍스트 분포도



출처: Xiaoyan Feng 외 4인, "Tracing Provenance and Detecting Tampering with Complementary LLM Watermarks", arXiv, 2026.08.13., https://arxiv.org/pdf/2608.12713

남은 한계도 존재한다. 편집이 텍스트 맨 뒤쪽에만 가해지면 취약 신호의 시드는 끝부분만 어긋나기 때문에 전체 점수의 하락 폭이 작아지며, 이를 해결하려면 뒷부분 구간을 따로 채점하는 방식이 필요하다. 또한 짧은 윈도를 쓰는 강건 신호는 질의를 반복해 묶음을 추정하는 신호 도용에 기존 기법과 같은 정도로 노출되어 있다. 단, 앞부분 전체를 시드로 삼는 취약 신호는 같은 방식으로 위조하기 어려워 위조된 텍스트는 '변조됨'으로 걸러질 수 있다. 이러한 점에서 두 신호를 나누어 배치하는 설계는 편승 위조에 대응하는 것을 넘어 도용 상황에서도 판정을 유지하는 효과를 낳는다.

III. 결론: 신뢰 가능한 워터마크 체계의 과제

• 출처 표시를 넘어선 변조 확인의 과제

하나의 워터마크로 출처와 변조를 함께 확인하려면 성질이 다른 두 신호를 하나의 텍스트에 겹쳐 심고, 두 점수를 순서대로 읽어 판정을 세 가지로 나누는 방식이 필요하다는 것이 이 기법의 결론이다. 이를 통해 워터마크는 산출물이 어느 모델에서 나왔는지를 답하는 데 그치지 않고, 전달된 내용이 생성 당시와 같은지를 함께 답한다. 이러한 점에서 워터마크는 출처를 표시하는 수단을 넘어 내용이 그대로 유지되었는지 확인하는 수단으로 확장되어, 산출물의 변형 여부를 확인할 수 있는 근거가 된다. 앞서 살펴본 클로드의 사례에서 제기된 논의, 즉 하나의 워터마크가 모델이 새로 만든 글과 다듬기만 한 글을 구분하지 못하는 문제도 워터마크가 담는 정보를 여러 축으로 나누는 방향에서 접근할 여지가 있다.

다만 실제 운영으로 이어지려면 남은 과제가 있다. 텍스트 뒷부분에만 가해진 편집은 취약 신호의 점수를 충분히 떨어뜨리지 못하므로 구간별 채점을 더하는 방식의 보완이 필요하고, 짧은 윈도를 쓰는 강건 신호가 신호 도용에 노출되는 문제도 남아 있다. 제도 측면에서는 워터마크가 무엇을 보증하는지를 명확히 정의하는 작업이 요구되며, 생성된 텍스트인지, 처리를 거친 텍스트인지, 전달 과정에서 변형된 텍스트인지를

구분해 표시하는 기준이 함께 마련될 필요가 있다. 따라서 산업계는 탐지 도구의 공개 범위와 검증 절차를 갖추고, 정규화 규칙처럼 서로 다른 서비스가 같은 결과를 내야 하는 부분에서는 공통 규격을 논의하는 방향으로 나아갈 것으로 보인다.

참고문헌

- Xiaoyan Feng 외 4인, "Tracing Provenance and Detecting Tampering with Complementary LLM Watermarks", arXiv, 2026.08.13., https://arxiv.org/pdf/2608.12713
- Ashley Belanger, "Claude's new Scarlet Letter watermark is invisible—for now", Ars Technica, 2026.08.13., https://arstechnica.com/tech-policy/2026/08/claudes-new-scarlet-letter-watermark-is-invisible-for-now/

기술용어

순번	용어	설명
1	C2PA (Coalition for Content Provenance and Authenticity)	콘텐츠의 출처와 제작 이력을 확인할 수 있는 공통 규격을 개발하기 위해 여러 기업과 기관이 결성한 연합체
2	**폭 없는 공백** (zero width)	화면에 글자나 공간을 차지하지 않으면서 단어의 경계나 줄 바꿈 위치를 지정하는 데 쓰이는 보이지 않는 특수 문자
3	**유니코드** (Unicode)	전 세계의 모든 문자를 컴퓨터에서 똑같이 표현하기 위해 만든 국제 표준 문자 체계
4	**동형 문자** (homoglyph)	모양이 아주 비슷하거나 똑같아서 눈으로 구별하기 어려운 글자. 대표적인 사례는 숫자 0과 대문자 O, 영어 알파벳 a와 키릴문자 a

인공지능 생성 텍스트 워터마크 기술의 발전과 향후 과제

뉴스 브리프

인공지능이 생성한 텍스트에 눈에 보이지 않는 워터마크를 심어 출처를 확인하는 기술이 확산하고 있다. 워터마크는 육안으로 볼 수 없지만 특정 방법으로는 판별할 수 있는 표식이다. 최근에는 이 워터마크를 지운다는 제거 도구가 웹 곳곳에 등장했으나, 판별 도구가 공개되지 않아 제거 효과를 입증한 사례는 드문 것으로 확인되었다. 특히 문장을 통째로 다시 쓰는 패러프레이징은 워터마크를 지우는 사실상 유일한 수단으로 지목되면서, 워터마크가 얼마나 견고하게 유지되는지가 쟁점으로 떠올랐다. 이에 워터마크 로직을 모델 내부에 직접 심어 패러프레이징에도 지워지지 않게 만드는 기술이 제시되었다. 본 보고서는 이러한 워터마크의 작동 방식과 패러프레이징 공격에 대한 대응 원리를 분석한다.

뉴스 플러스

Ⅰ. 서론: 인공지능 워터마크를 둘러싼 논쟁

- **워터마크 제거 도구의 확산과 검증 문제**

2026년 8월, 미국의 AI 개발기업 앤트로픽(Anthropic)은 자사의 인공지능 서비스 클로드 (Claude)가 생성하는 모든 텍스트에 눈에 보이지 않는 워터마크를 심기 시작했다. 워터마크는 사람의 눈에는 드러나지 않지만 특정 방법으로는 판별할 수 있는 표식으로, 해당 텍스트가 인공지능이 만든 것인지 확인하는 수단이다. 이 사실이 알려진 지 며칠 만에, 워터마크를 지워 준다고 내세우는 제거 도구가 웹 곳곳에 잇따라 등장했다.[7] 개발자들이 코드를 공유하고 협업하는 공간인 깃허브(GitHub)에서 관심 표시를 4,500개 넘게 받은 프로젝트부터 새로 개설된 여러 웹 도구, 기존의 인공지능 탐지 우회 서비스까지 그 형태도 다양하다.

7) Ax Sharma, “AI 'watermark removers' flood the web. Almost none can prove they work.”, Bleeping Computer, 2026.08.13., https://www.bleepingcomputer.com/news/security/ai-watermark-removers-flood-the-web-almost-none-can-prove-they-work/

그러나 이들 도구가 광고하는 대로 워터마크를 실제로 제거하는지는 외부에서 확인하기 어려운 상황이다. 앤트로픽이 워터마크의 작동 방식과 이를 판별하는 탐지기를 공개하지 않아, 제거를 마쳤다는 문서가 여전히 워터마크를 지니고 있는지 확인할 근거가 없기 때문이다.[8] 제거 도구가 다루는 대상은 크게 두 갈래로 나뉘는데, 텍스트에 섞인 보이지 않는 특수 문자나 파일에 딸린 부가 정보를 지우는 일은 실제로 가능하고 그 결과도 눈으로 확인할 수 있다. 다만 이는 손쉽게 지워지는 표층적 흔적에 그치며, 화면을 갈무리하거나 파일을 다시 저장하는 것만으로도 사라지는 수준이다. 정작 핵심 신호는 모델이 어떤 단어를 골랐는가라는 통계적 패턴에 담겨 있어, 이를 지우려면 다른 모델로 문장을 통째로 다시 쓰는 패러프레이징(paraphrasing) 방식이 사실상 유일한 방법으로 지목된다.

• 오픈소스 환경과 워터마크 내장의 필요성

워터마크를 심는 방식은 크게 두 갈래로 나뉜다. 하나는 모델이 문장을 만들어 내는 순간에 개입해 특정 단어가 더 자주 선택되도록 유도하는 방식이다. 이 방식은 널리 쓰이지만, 모델의 내부를 누구나 열어 볼 수 있는 오픈소스(open-source) 환경에서는 한계가 뚜렷하다. 오픈소스 모델은 이용자가 생성 과정을 직접 통제할 수 있어, 워터마크를 심는 절차를 손쉽게 꺼 버릴 수 있기 때문이다. 앞서 살펴본 제거 도구가 겨냥하는 대상 가운데 하나도 이렇게 생성 과정에 의존하는 워터마크다.

이러한 한계는 워터마크를 모델의 겉이 아니라 안에 심어야 한다는 문제로 이어진다. 생성 과정에 얹는 장치가 아니라 모델의 가중치(weight) 자체에 워터마크를 심으면, 이용자가 이를 임의로 제거하기 어려워진다. 최근 제시된 오픈스탬프(OpenStamp)는 이러한 방향을 구체화한 기술로, 모델이 단어마다 점수를 매기는 마지막 층에 워터마크를 심는다. 워터마크를 지우려는 시도가 이어지는 상황에서, 쉽게 제거되지 않는 워터마크를 모델에 심는 일은 창작물의 출처를 확인하려는 기술적 대응의 출발점이 된다.

II. 본론: 오픈스탬프 워터마크의 작동 원리

• 언임베딩 층 수정과 그린 리스트 편향

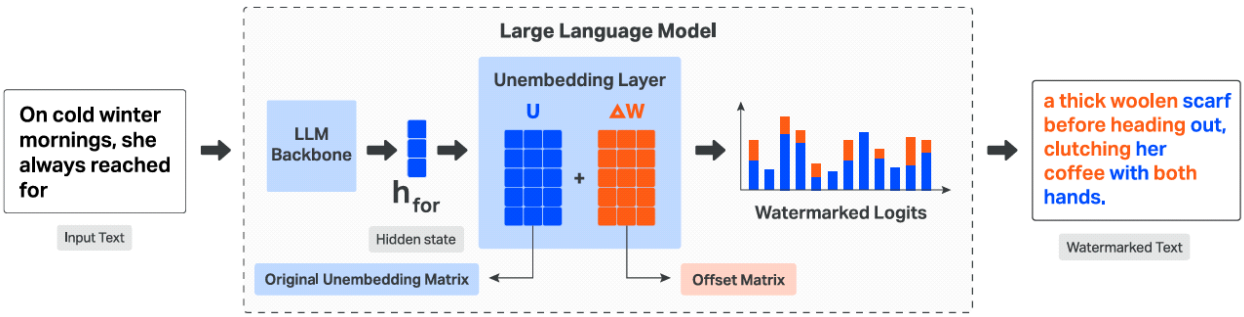
오픈스탬프를 이해하려면 먼저 언어 모델이 단어를 골라내는 마지막 단계를 살펴봐야 한다. 모델은 문장을 이어 갈 때 내부에서 계산한 값을 언임베딩 층(unembedding layer)이라는 마지막 층에 통과시킨다. 언임베딩 층은 모델이 계산한 내부 값을 어휘 사전에 있는 각 단어의 점수로 변환해주는 역할을 수행하며, 이 점수가 높은 단어일수록 다음 단어로 선택될 확률이 커진다. 오픈스탬프는 바로 이 언임베딩 층의 값을 조정해 특정 단어들이 조금 더 높은 점수를 받도록 만든다. 이 미세한 편향이 문장 전체에 쌓이면서, 사람 눈에는 드러나지 않지만 판별은 가능한 워터마크가 심어진다.

8) Ax Sharma, “AI 'watermark removers' flood the web. Almost none can prove they work.”, Bleeping Computer, 2026.08.13., https://www.bleepingcomputer.com/news/security/ai-watermark-removers-flood-the-web-almost-none-can-prove-they-work/

이때 어떤 단어에 점수를 더 줄지 정하는 기준이 그린 리스트(green list)다. 그린 리스트는 전체 어휘 가운데 점수를 높여 줄 단어들을 선별한 목록으로, 그린 리스트에 든 단어가 더 자주 등장하도록 유도하는 것이 이 기술의 핵심이다. 그린 리스트에 든 단어의 비율과 점수는 조절할 수 있으며, 워터마크를 강하게 심을수록 판별은 쉬워지지만 문장의 자연스러움은 줄어든다. 반대로 약하게 심으면 문장은 매끄럽지만 판별이 어려워진다. 워터마크의 강도와 문장의 품질 사이에서 균형을 맞추는 일이 관건이 된다.

다만 그린 리스트를 고정해 두면 문제가 생긴다. 늘 같은 단어들이 밀어 올려지면, 어떤 단어가 그린 리스트에 있는지 역으로 추측당하기 쉽고, 그린 리스트가 파악되면 워터마크는 무력해진다. 오픈스탬프는 이를 피하기 위해 여러 개의 그린 리스트를 두고, 문장의 맥락에 따라 매번 다른 그린 리스트를 골라 쓴다. 어떤 맥락에서 어떤 그린 리스트가 쓰일지 예측하기 어려워지므로, 그린 리스트를 알아내 워터마크를 지우려는 시도를 막는 데 도움이 된다. 이렇게 맥락에 따라 그린 리스트를 바꾸는 방식은 다음 소제목에서 다룰 패러프레이징 대응의 토대가 된다.

[그림 1] 오픈스탬프의 워터마크 삽입 과정



출처: Miroojin Bakshi 외 2명, “OpenStamp: A Watermark for Open-Source Language Models”, arXiv, 2026.08.28., https://arxiv.org/pdf/2608.27899

• 패러프레이징 공격에 대응하는 의미 정렬

오픈스탬프가 겨냥하는 가장 큰 위협은 패러프레이징이다. 패러프레이징은 문장의 뜻은 그대로 두고 단어와 문장 구조만 바꿔 다시 쓰는 것을 말한다. 워터마크가 특정 단어의 선택에 담겨 있다 보니, 다른 모델로 문장을 통째로 다시 쓰면 그 단어들이 대부분 교체되어 워터마크가 흐려진다. 앞서 살펴본 제거 도구가 워터마크를 지우는 사실상 유일한 방법으로 패러프레이징을 지목한 것도 이 때문이다. 따라서 패러프레이징을 거쳐도 워터마크가 남아 있게 만드는 일이 오픈스탬프의 핵심 과제가 된다.

오픈스탬프는 이를 위해 뜻이 비슷한 문장이라면 단어가 달라도 같은 그린 리스트가 선택되도록 설계한다. 그 열쇠가 투영(projection)이라는 절차다. 투영은 모델이 계산한 내부 값을 뜻이 비슷한 문장끼리 가깝게 모이는 공간으로 옮기는 과정으로, 원래의 내부 값은 다음 단어를 맞히는 데 맞춰져 있어 뜻의 유사성을

그대로 반영하지 못하기 때문에 필요하다. 뜻이 가까운 문장들이 이 공간에서 서로 가깝게 모이면, 같은 맥락으로 묶여 같은 그린 리스트를 고르게 된다. 그 결과 원래 문장과 패러프레이징한 문장이 단어는 달라도 같은 그린 리스트로 이어져, 워터마크가 이어진다.

이 설계의 효과는 실험으로도 확인된다. 투영 절차를 넣은 오픈스탬프는 패러프레이징을 거친 문장에서도 다른 방법들보다 높은 판별 성능을 보였다. 반대로 투영 절차를 빼면 성능이 눈에 띄게 떨어져, 이 절차가 패러프레이징 대응에 실제로 기여함이 드러났다. 이는 워터마크를 단어 하나하나가 아니라 문장의 뜻에 걸어 두었기 때문으로 해석된다. 단어를 바꾸는 공격으로는 의미에 심겨진 워터마크까지 지우기 어렵다는 점에서, 패러프레이징으로 워터마크를 무력화하려는 시도에 대한 기술적 방어선이 된다.

• 로그 우도비를 활용한 워터마크 탐지

워터마크를 심었다면 그다음은 어떤 문장에 워터마크가 들어 있는지 탐지하는 일이다. 오픈스탬프는 이를 위해 두 개의 모델을 나란히 놓고 비교한다. 하나는 워터마크를 심은 모델이고, 다른 하나는 워터마크를 심지 않은 원래 모델이다. 같은 문장을 두 모델에 각각 넣어, 그 문장이 나올 법한 정도를 모델별로 계산한 뒤 서로 비교하는 것이다. 워터마크를 심은 모델은 그린 리스트에 든 단어가 더 자주 나오도록 편향되어 있으므로, 실제로 워터마크가 들어 있는 문장이라면 워터마크를 심은 모델 쪽에서 더 나올 법한 문장으로 계산된다.

두 모델의 확률을 비교하여 도출한 값이 로그 우도비(log-likelihood ratio)다. 로그 우도비는 어떤 문장이 워터마크를 심은 모델에서 나왔을 가능성이 원래 모델에서 나왔을 가능성보다 얼마나 큰지를 하나의 수치로 나타낸 값이다. 이 값이 미리 정한 기준을 넘으면 워터마크가 들어 있는 문장으로 판정한다. 문장 길이로 값을 고르게 맞추기 때문에 짧은 문장과 긴 문장에 같은 기준을 적용할 수 있고, 문장을 만들어 낼 때 쓴 앞부분을 몰라도 탐지가 가능하다. 다만 이 방식은 워터마크를 심은 모델과 원래 모델의 값을 모두 계산해야 하므로, 문장만 통계적으로 훑어보는 방법보다 탐지에 드는 계산 부담이 크다.

• 사후 변형과 제거 시도에 대한 견고성

워터마크가 실제로 쓸모가 있으려면, 모델이 공개된 뒤 이용자가 손을 대더라도 워터마크가 남아 있어야 한다. 오픈소스 모델은 공개 이후 여러 방식으로 변형되기 때문이다. 대표적인 것이 미세조정(fine-tuning)이다. 미세조정은 이미 학습을 마친 모델에 새로운 데이터를 추가로 학습시켜 성능이나 용도를 조정하는 과정으로, 이 과정에서 모델의 값이 바뀌면 워터마크도 함께 흐려질 수 있다. 실험 결과, 오픈스탬프도 미세조정이 진행될수록 탐지 성능이 떨어졌으나, 그 하락 폭이 다른 방법들보다 작아 더 높은 탐지 성능을 유지했다.

이용자가 흔히 적용하는 다른 변형에도 워터마크는 견뎠다. 그 하나가 양자화(quantization)다. 양자화는 모델이 쓰는 수치의 정밀도를 낮춰 용량과 계산 비용을 줄이는 방법으로, 성능을 크게 해치지 않으면서 모델을 가볍게 만들 때 널리 쓰인다. 오픈스탬프는 양자화를 거친 뒤에도 탐지 성능이 거의 떨어지지 않았다.

워터마크를 지우려고 언임베딩 층을 통째로 초기화한 뒤 다시 학습시키는 시도도 검토되었는데, 이 경우 워터마크는 지워지더라도 문장 품질을 원래 수준으로 되돌리기가 어려워, 상당한 계산 비용을 치러야 했다.

이러한 견고성은 앞서 살펴본 제거 도구의 확산과 곧바로 맞닿는다. 생성 과정에 얹은 워터마크는 손쉽게 꺼 버릴 수 있지만, 모델의 값 자체에 심은 워터마크는 미세조정이나 양자화 같은 변형을 거쳐도 쉽게 사라지지 않는다. 물론 오픈스탬프에도 한계는 있다. 문장이 아니라 명령을 따르는 형태의 출력에서는 탐지 성능이 떨어졌고, 탐지에 원래 모델이 필요해 계산 부담이 크다. 그럼에도 워터마크를 모델 안에 심는 방식은 지우려는 시도에 견디는 힘을 갖췄다는 점에서, 창작물의 출처를 확인하려는 기술적 대응을 한 걸음 진전시킨 것으로 평가된다.

III. 결론: 텍스트 워터마크 기술의 향후 과제

• 기술과 제도의 균형점 모색

인공지능이 만든 텍스트가 늘어나면서, 그 출처를 확인하려는 기술과 이를 지우려는 시도가 맞서고 있다. 워터마크를 지운다는 제거 도구가 나오고 있지만, 손쉽게 지워지는 것은 표층의 흔적일 뿐이고 정작 핵심 신호는 단어 선택에 담겨 있어 문장을 통째로 다시 쓰지 않는 한 지우기 어렵다. 오픈스탬프는 이 핵심 신호를 모델의 값 자체에 심고, 뜻이 비슷한 문장이라면 같은 그린 리스트가 선택되도록 설계해 패러프레이징에도 워터마크가 남도록 만들었다. 결국 이는 워터마크를 단어의 겉이 아니라 문장의 뜻에 심어 두었을 때 비로소 지우기 어려운 워터마크가 된다는 점을 의미한다. 이러한 점에서 오픈스탬프는 창작물의 출처를 확인하려는 기술이 제거 시도에 맞서 한 걸음 나아간 사례로 평가된다.

다만 기술만으로 문제가 풀리지는 않는다. 오픈스탬프는 명령을 따르는 형태의 출력에서 탐지 성능이 떨어지고, 탐지에 원래 모델이 필요해 계산 부담이 크다는 한계를 안고 있다. 따라서 워터마크가 더 다양한 상황에서 견디도록 기술을 다듬는 일과 함께, 탐지 결과를 신뢰할 수 있는 근거로 삼을 제도적 기준을 마련하는 일이 필요하다. 워터마크를 심는 쪽과 탐지하는 쪽이 같은 규칙을 공유하고, 그 결과를 누구나 확인할 수 있도록 절차를 투명하게 공개하는 일도 과제로 남는다. 이를 통해 인공지능 기술의 발전과 창작물의 출처를 확인하려는 요구가 함께 나아갈 토대가 마련될 것으로 보인다.

참고문헌

• Miroojin Bakshi 외 2명, “OpenStamp: A Watermark for Open-Source Language Models”, arXiv, 2026.08.28., https://arxiv.org/pdf/2608.27899
• Ax Sharma, “AI 'watermark removers' flood the web. Almost none can prove they work.”, Bleeping Computer, 2026.08.13., https://www.bleepingcomputer.com/news/security/ai-watermark-removers-flood-the-web-almost-none-can-prove-they-work/