The Files are in the Computer: On Copyright, Memorization, and Generative AI

Chicago-Kent Law Review (forthcoming 2024)

A. Feder Cooper*
James Grimmelmann†

The New York Times*'s copyright lawsuit against OpenAI and Microsoft alleges that OpenAI's GPT models have "memorized"* Times *articles. Other lawsuits make similar claims. But parties, courts, and scholars disagree on what memorization is, whether it is taking place, and what its copyright implications are. Unfortunately, these debates are clouded by deep ambiguities over the nature of "memorization," leading participants to talk past one another.*

In this Essay, we attempt to bring clarity to the conversation over memorization and its relationship to copyright law. Memorization is a highly active area of research in machine learning, and we draw on that literature to provide a firm technical foundation for legal discussions. The core of the Essay is a precise definition of memorization for a legal audience. We say that a model has "memorized" a piece of training data when (1) it is possible to reconstruct *from the model (2) a* near-exact copy *of (3) a* substantial portion *of (4)* that specific piece of training data. *We distinguish memorization from "extraction" (in which a user intentionally causes a model to generate a near-exact copy), from "regurgitation" (in which a model generates a near-exact copy, regardless of the user's intentions), and from "reconstruction" (in which the near-exact copy can be obtained from the model by any means, not necessarily the ordinary generation process).*

Several important consequences follow from these definitions. First, not all learning is memorization: much of what generative-AI models do involves generalizing from large amounts of training data, not just memorizing individual pieces of it. Second, memorization occurs when a model is trained; it is

*. Co-Founder, The GenLaw Center; Assistant Professor of Computer Science, Yale University (to commence 2026). After completion of this Essay (but prior to its official publication), A. Feder Cooper started as a Postdoctoral Researcher at Microsoft Research and a Postdoctoral Affiliate at Stanford University. Both authors contributed equally to this Essay. We presented an earlier version of this Essay at the Chicago-Kent Law Review AI Disrupting Law Symposium on April 26, 2024. Our thanks to the organizers and participants, and to Aislinn Black, Derek Bambauer, Annmarie Bridy, Fernando Delgado, Aaron Gokaslan, Katherine Lee, Colin Raffel, Matthew Sag, Benjamin Sobel, Pamela Samuelson, Jessica Silbey, and Eugene Volokh.

†. Tessler Family Professor of Digital and Information Law, Cornell Tech and Cornell Law School; Researcher, The GenLaw Center.

not something that happens when a model generates a regurgitated output. Regurgitation is a symptom of memorization in the model, not its cause. Third, when a model has memorized training data, the model is a "copy" of that training data in the sense used by copyright law. Fourth, a model is not like a VCR or other general-purpose copying technology; it is better at generating some types of outputs (possibly including regurgitated ones) than others. Fifth, memorization is not just a phenomenon that is caused by "adversarial" users bent on extraction; it is a capability that is latent in the model itself. Sixth, the amount of training data that a model memorizes is a consequence of choices made in the training process; different decisions about what data to train on and how to train on it can affect what the model memorizes. Seventh, system design choices also matter at generation time. Whether or not a model that has memorized training data actually regurgitates that data depends on the design of the overall system: developers can use other guardrails to prevent extraction and regurgitation. In a very real sense, memorized training data is in the model—to quote Zoolander*, the files are in the computer.*

> *Matilda*: Did you find the files?
> *Hansel*: I don't even know what they—what do they look like?
> *Matilda*: They're in the computer.
> *Hansel*: They're in the computer?
> *Matilda*: Yes, they're definitely in there, I just don't know how he labeled them.
> *Hansel*: I got it. IN the computer. It's so simple.[1]

I. Introduction

The week between Christmas and New Year's Eve is usually a slow news week, but not in 2023, the year that ChatGPT ate the world.[2] On December 27, *The New York Times* filed a massive copyright-infringement lawsuit against Microsoft and OpenAI, alleging that Bing Copilot and ChatGPT constituted "massive copyright infringement."[3] In particular, the *Times* alleged that the generative-AI models in these systems had "memorized" large quantities of *Times* articles. When prompted with some prefix of text from a *Times* article,[4] ChatGPT would output a lengthy, corresponding suffix that copied passages from the article—hundreds of words, varying only in a few scattered portions.[5] (See Figure 1.)

To the *Times* and its lawyers, these examples of "memorization" were blatant copyright infringement. But to OpenAI and its defenders, there was nothing to see here. OpenAI responded, both in court and online, that these examples were "adversarial, not normal usage patterns." On this view, any copying (and thus any resulting infringement) resulted from the prompts the *Times* used. If the *Times* had not specifically manipulated ChatGPT into generating *Times* articles, there would have been no copying, and no copyright infringement. As economist Tyler Cowen put it, in mocking the *Times*'s argument, one could equally well say that a toothpick infringes:

> If you stare at just the exact right part of the toothpick, and measure the length from the tip, expressed in terms of the appro-

1. Zoolander (Paramount Pictures 2001).

2. *See generally Chat GPT Is Eating the World* (2024), https://chatgptiseatingtheworld.com.

3. Complaint at ¶ 74, N.Y. Times Co. v. Microsoft, No. 2:24-cv-00711 (C.D. Cal. Dec. 27, 2023).

4. The prompts ranged in length from a sentence to several paragraphs. *See id.* Exh. J.

5. At this point, in the introduction, we are deliberately providing only a high-level intuition for the type of memorization with which we engage in this Essay, and we do so by drawing on an example from a current lawsuit. We provide more rigorous definitions of memorization below. *See infra* notes 51–55 and accompanying text.

Output from GPT-4:

exempted it from regulations, subsidized its operations and promoted its practices, records and interviews showed.

Their actions turned one of the best-known symbols of New York — its yellow cabs — into a financial trap for thousands of immigrant drivers. More than 950 have filed for bankruptcy, according to a Times analysis of court records, and many more struggle to stay afloat.

"Nobody wanted to upset the industry," said David Klahr, who from 2007 to 2016 held several management posts at the Taxi and Limousine Commission, the city agency that oversees medallions. "Nobody wanted to kill the golden goose."

New York City in particular failed the taxi industry, The Times found. Two former mayors, Rudolph W. Giuliani and Michael R. Bloomberg, placed political allies inside the Taxi and Limousine Commission and directed it to sell medallions to help them balance budgets and fund key initiatives.

During that period, much like in the mortgage lending crisis, a group of industry leaders enriched themselves by artificially inflating medallion prices. They encouraged medallion buyers to borrow as much as possible and ensnared them in interest-only loans and other one-sided deals that often required borrowers to pay hefty fees, forfeit their legal rights and give up most of their monthly incomes.

When the market collapsed, the government largely abandoned the drivers who bore the brunt of the crisis. Officials did not bail out borrowers or persuade banks to soften loan

Actual text from NYTimes:

exempted it from regulations, subsidized its operations and promoted its practices, records and interviews showed.

Their actions turned one of the best-known symbols of New York — its signature yellow cabs — into a financial trap for thousands of immigrant drivers. More than 950 have filed for bankruptcy, according to a Times analysis of court records, and many more struggle to stay afloat.

"Nobody wanted to upset the industry," said David Klahr, who from 2007 to 2016 held several management posts at the Taxi and Limousine Commission, the city agency that oversees cabs. "Nobody wanted to kill the golden goose."

New York City in particular failed the taxi industry, The Times found. Two former mayors, Rudolph W. Giuliani and Michael R. Bloomberg, placed political allies inside the Taxi and Limousine Commission and directed it to sell medallions to help them balance budgets and fund priorities. Mayor Bill de Blasio continued the policies.

Under Mr. Bloomberg and Mr. de Blasio, the city made more than $855 million by selling taxi medallions and collecting taxes on private sales, according to the city.

But during that period, much like in the mortgage lending crisis, a group of industry leaders enriched themselves by artificially inflating medallion prices. They encouraged medallion buyers to borrow as much as possible and ensnared them in interest-only loans and other one-sided deals that often required them to pay hefty fees, forfeit their legal rights and give up most of their monthly incomes.

Figure 1: Memorized output from ChatGPT's GPT-4 endpoint (left) of a *New York Times* article (right)

> priate unit and converted into binary, and then translated into English, you can find any message you want. You just have to pinpoint your gaze very very exactly (I call this "a prompt").
>
> In fact, on your toothpick you can find the lead article from today's *New York Times*. With enough squinting, measuring, and translating.
>
> By producing the toothpick, they put the message there and thus they gave you NYT access, even though you are not a paid subscriber. You simply need to know how to stare (and translate), or in other words how to prompt.
>
> So let's sue the toothpick company![6]

6. Tyler Cowen, *Toothpick producers violate NYT copyright* (2023), https://marginalrevolution.com/marginalrevolution/2023/12/toothpick-producers-violate-nyt-copyright.html.

Implicit in this view is that memorization and the copying it involves take place only at *generation time*: when a generative-AI system responds to a user's prompt with an output. The system itself is a neutral, general-purpose tool. Some users may use it to infringe, but other users will not.

This view treats the machine-learned model (or models) at the heart of a generative-AI system as a black box. Training data is used to design and construct the box, but the box itself contains only abstracted statistical patterns of the training data. Those patterns either contain no expression at all, or if they do, they are represented in a way that is fundamentally uninterpretable. The box is a machine that transforms prompts into outputs. Thus, if there is infringing expression in the output, it must be because the user prompted it in a targeted (i.e., "adversarial" or "abnormal") way to elicit that infringement.

This view refuses to consider what happens inside the box—the specifics of *how* statistical learning about the training data enables generative-AI systems to do what they do. It avoids engaging with the actual representation of information about training data in a model's parameters. In legal writing, this has involved gesturing at these representations with high-level terms like "features," "patterns," or "statistical correlations."[7] These terms suggest that while there may be some underlying math going on, the details can be sidestepped for simplicity, because they are irrelevant to the legal treatment of Generative AI.

This way of thinking about memorization[8] has significant copyright consequences. It suggests that memorization is primarily about *prompting* rather than *training*. Outputs may contain infringing expression, but the model that generates them does not. A model itself is a neutral tool, equally good at producing infringing and non-infringing outputs. It follows that users bear most or all of the responsibility for misusing a generative-AI system to elicit memorized content, and the creators of the system in which the model is deployed bear little or none.[9]

---

7. *See, e.g.,* Oren Bracha, The Work of Copyright in the Age of Machine Production (Jan. 2024) (unpublished manuscript), https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4581738; Response, Concord Music Grp., Inc. v. Anthropic PBC, No. 3:23-cv-01092 (M.D. Tenn. Jan. 16, 2024).

8. For now, we continue to limit our use of the term "memorization" to the intuition provided in the (near-)verbatim copying demonstrated in Figure 1. *See infra* notes 51–55 and accompanying text (providing more details on memorization and variations on definitions).

9. The distinction between a *model* and the larger *system* in which it is embedded is important to keep in mind. *See infra* Part II.B (discussing technical difference). *See infra* Part III.I (discussing legal consequences).

With respect, we believe that this approach to making sense of memorization misdescribes how generative-AI systems work. If a generative-AI model memorizes its training data, the training data is *in the model*. This should not be surprising. Models are not inert tools that have no relationship with their training data. The power of a model is precisely that it encodes relevant features of the training data in a way that enables prompting to generate outputs that are based on the training data. This is why capital-G Generative AI is such a big deal. All useful models learn something about their training data. Memorization is simply a difference in degree: it is an encoded feature *in the model*; whether it is a desired feature or not is another matter entirely.

It follows that memorization in Generative AI cannot be neatly confined to generation time—to how the system behaves when adversarial users provide adversarial prompts. If a generative-AI model has memorized copyrighted works, the memorized aspects of those works are present *in the model itself*, not just in the model's generated outputs. The model can possibly generate copies of those works on demand for any user,[10] not just for users who have a suitably nefarious intent. The system's creator may have various options to limit infringing outputs—for example, by refusing to generate outputs for certain prompts, or by checking outputs against a database of copyrighted works before returning them to the user. But one of these options is always to *change the model*: to train or retrain it in a way that attempts to limit the model's memorization of training data. Whether this is trivially easy or impractically hard depends on the details of the model architecture, the choice of training data, the training algorithm, and much more. But the model's internals must always be part of the technical picture, because they are highly relevant to what a model has memorized and what it can do.

We take no position on what the most appropriate copyright regimes for generative-AI systems should be, and we express no opinion on how pending copyright lawsuits should be decided. These cases raise difficult doctrinal issues that run almost the entire gamut of copyright law.[11] Our goal is merely to describe how these systems work so that copyright scholars can develop their theories of Generative AI on a firm technical foundation. We focus on a few threshold issues—particularly the Copyright Act's definition of "copies"—where the technical details are particularly salient. We seek clarity, precision, and technical accuracy.

---

10. With some probability.

11. *See generally* Katherine Lee, A. Feder Cooper & James Grimmelmann, *Talkin' 'Bout AI Generation: Copyright and the Generative-AI Supply Chain*, Journal of the Copyright Society of the U.S.A (forthcoming) [hereinafter Talkin'], https://arxiv.org/abs/2309.08133v2.

You have nearly finished reading Part I of this Essay, the introduction. In Part II, we provide a brief background on how generative-AI models work, and the systems and supply chains within which they are embedded. In Part III, the heart of the Essay, we describe how to think clearly about memorization in generative-AI systems, and show how several common arguments about copyright and Generative AI are built on a mistaken view of what memorization consists of and how it is surfaced to end users. Part IV offers a brief conclusion, with some historical reflections.

II. Technical Background

In the past year and a half—starting roughly with the public launch of ChatGPT in November 2022—Generative AI has become a household term. It is used as a blanket description for a wide range of consumer-facing applications: chatbots like OpenAI's ChatGPT Plus,[12] Google DeepMind's Gemini,[13] and Anthropic's Claude 3;[14] image generators like Midjourney Inc.'s epopynmous Midjourney,[15] StabilityAI's Stable Diffusion,[16] and OpenAI's DALL·E-3;[17] music generators like Google DeepMind's Lyria;[18] video generators like Pika's eponymous Pika[19] and OpenAI's Sora[20]; programming assistants like GitHub Copilot; and much more. These tools are self-evidently different from one another. They operate on different data *modalities* (text, image, audio, video, and code, respectively),[21] incorporate different types of

12. *DALL·E 3 is now available in ChatGPT Plus and Enterprise*, OpenAI (Oct. 19, 2023), https://openai.com/blog/dall-e-3-is-now-available-in-chatgpt-plus-and-enterprise.

13. Gemini Team et al., Gemini: A Family of Highly Capable Multimodal Models (2023) (unpublished manuscript), https://arxiv.org/abs/2312.11805.

14. Anthropic, *Introducing the next generation of Claude* (Mar. 4, 2024), https://www.anthropic.com/news/claude-3-family.

15. *Midjourney* (2023), https://midjourney.com/.

16. Robin Rombach, Andreas Blattmann & Dominik Lorenz et al., *High-Resolution Image Synthesis with Latent Diffusion Models*, *in* 2022 2022 IEEE Conf. on Comput. Vision & Pattern Recognition (2022); *Stable Diffusion XL*, Stability AI (2023), https://stability.ai/stablediffusion.

17. OpenAI, *DALL·E 3* (2023), https://openai.com/dall-e-3; James Betker, Gabriel Goh & Li Jing et al., Improving Image Generation with Better Captions (2023) (unpublished manuscript), https://cdn.openai.com/papers/dall-e-3.pdf.

18. Google DeepMind, *Transforming the future of music creation* (Nov. 16, 2023), https://deepmind.google/discover/blog/transforming-the-future-of-music-creation/.

19. Pika, *An idea-to-video platform that brings your creativity to motion* (2023), https://pika.art/.

20. OpenAI, *Creating video from text* (2024), https://openai.com/sora.

21. Talkin', *supra* note 11, at 18–24 (defining and describing modalities).

model architectures, interact with different software-systems components, are made available in different ways, and serve different purposes.

But beneath their differences, these Generative AI tools have a common shape that justifies the use of the same term to describe them all. This part describes that common shape. Section A presents the (highly simplified) basics of deep-neural-network machine learning that powers most modern generative-AI models. Section B describes the supply chains in which generative-AI models are embedded—supply chains that connect data to models to usable systems to outputs.

A. Generative AI

First, Generative AI involves *machine-learning models* that have been created through *training* on *datasets* that contain massive numbers of data *examples*.[22] Second, these models are all *generative*: they produce outputs of the same modality as their training data.[23]

This second point is what distinguishes generative-AI models from other ML models. A classifier (a type of *discriminative* model) will typically be trained on information-rich *training examples*, such as a collection of JPEG images of cats and dogs. When the trained classifier is run on an new JPEG, it will output either a simple label of `cat` or `dog`, based on whether it predicts that the JPEG is more likely to be an image of a cat or an image of a dog.

In contrast, while generative-AI models are also trained on information-rich training examples, their outputs are (1) also information-rich and (2) of the same type as their training examples.[24] A generative image model, for example, might be trained on images and their captions; after trained, it can then take a text input (e.g., `"cat in a red and white striped hat"`), and produce as output one of many possible different images of cats in red and white striped hats.[25]

---

22. *See generally id.* at 24–30.

23. Some models are *multimodal*: they are trained on multiple modalities and, for example, take one modality as input and produce another as output. This is the case for text-to-image generation models like Stable Diffusion. Stable Diffusion is trained on image-caption pairs; it takes text prompts as inputs and produces image generations as outputs. *See* Rombach, Blattmann & Lorenz et al., *supra* note 16 (discussing the original Stable Diffusion training process).

24. In general, what constitutes a single training example varies across models, and examples do not necessarily cleanly map to complete creative works. Consider the text modality: a single training example may be a piece of one long work, which has been broken up into pieces and spread across multiple examples.

25. This example is drawn from Talkin', *supra* note 11. *See id.* at 8–15 (providing more extensive background on generative modeling in comparison to discriminative modeling).

In a bit more detail, the objective of training is to create a generative-AI model that produces outputs that reflect patterns in the training data.[26] This coheres with copyright-lawsuit defendants' own descriptions of the training process and resulting trained models. For example:

> . . . [During training,] AI models like Claude ingest billions of different kinds of texts, which they break down into trillions of component parts known as "tokens." The models then analyze the "tokens to discern statistical correlations—often at staggeringly large scales—among features of the content on which the model is being trained." Those statistical correlations effectively yield "insights about patterns of connections among concepts or how works of [a particular] kind are constructed." Based on those insights, models like Claude are able to create new, original outputs with a degree of sophistication and verisimilitude that approximates human cognition.[27]

The model-training process is fundamentally statistical: it learns statistics about the training data. Each training example is regarded as a sample from a *distribution* of possible examples—e.g., each picture of a cat in the training set is one sample drawn from the hypothetical space of possible pictures of cats. A training algorithm attempts to learn the distribution from which the training examples are drawn. If training is successful, then the model's outputs (generated images from the hypothetical learned distribution of images of cats) will share statistical properties with actual images drawn from the actual real-life distribution of images of cats from which the training examples were taken. In other words, we can think of generative-AI models as *ML models that produce outputs that exhibit statistical properties derived from the examples on which they were trained*.

This summary shows both how phrases like "pattern" and "statistical correlation" are useful abstractions for understanding model training, and also the limits of these abstractions. Such "statistical correlations" can encompass many different things in training data. In an image model, they can be concepts (e.g., a cat as being a furry, tailed, four-legged animal), styles

26. The *goal* of training is different from this underlying mathematical *objective*. The overarching goal is to produce useful or delightful models, which is not exactly the same as the mathematical objective used to train these models. *See* A. Feder Cooper, Katherine Lee, James Grimmelmann & Daphne Ippolito et al., Report of the 1st Workshop on Generative AI and Law 4 (2023), https://arxiv.org/abs/2311.06477 (discussing this distinction).

27. Response at 4–5, Concord Music Grp., Inc. v. Anthropic PBC, No. 3:23-cv-01092 (M.D. Tenn. Jan. 16, 2024) (internal citations omitted). *See infra* Part III.A (for additional discussion of this quote in the context of memorization).

(e.g., photorealism), artistic media (e.g., oil painting), and more. At generation time these elements can be remixed to produce new images that never existed before and that are highly dissimilar from all examples in the training dataset (e.g., a cat in a red and white striped hat). Sometimes, they can also be "re"-mixed to (re-)produce particular training examples: anything can be described perfectly in terms of "patterns" and "statistical correlations" if they are detailed enough.

There is even more complexity in practice. There are many different types of generative-AI models, which have radically different technical architectures. But, at a high (and over-simplified) level of abstraction, they generally consist of *neural networks*: interconnected nodes that can perform computations, and which are organized into layers. The strengths of these connections—the influences that nodes have on another—is what is learned during training. These are called the model *parameters* or *weights*, and they are represented as numbers.

To run a generative-AI model on an input—a *prompt*—a computer program takes the prompt and transforms it into a format that can be processed in the model. For large language models (LLMs), this typically involves taking the prompt and converting it into *tokens* (words or parts of words, as described above).[28] The transformed, tokenized prompt is passed through the layers of the neural network: the computer program copies the input into the nodes at the first layer of the network, then uses the parameters (i.e., connection strengths) leading out from those nodes to compute the input's effects on the nodes in the second layer, and so on, until the last layer has been computed. For example, in LLMs, this process determines how important each token is in relation to the entire sequence of tokens that make up the text prompt.[29]

At this point, once the prompt has been processed through all of the model's layers, the model will produce an output. For LLMs, this means the model will predict the most likely next token in the sequence, based on the context of the prompt, and *generate* that token as the next token in the se-

28. These tokens represent whole words or parts of words, and are the format that the model can process directly. These tokens are then mapped to embedding vectors, which reflect underlying semantic and syntactic information about the words they encode. *Id.* (discussing tokenization at a high level); Talkin', *supra* note 11, at 10–15 (and citations therein); Vicki Boykis, What are embeddings? (June 2023), https://github.com/veekaybee/what_are_embeddings (for an accessible treatment of the details behind embeddings).

29. Diffusion-based image generation models typically also involve neural networks, but also undergo a different training process. We omit these details in this Essay.

quence.[30] What is "most likely" depends on the "statistical correlations"[31] learned during training. For example, if trained on a dataset that includes fairy tales, a model would (probably) deem `"time"` the most likely next token to follow `"once upon a"`.[32] In practice, the generation process tends to be iterative: once a token is generated, it is appended to the prompt, and the new, extended prompt is provided as input to the model, which generates the next token in the sequence.

Generative modeling has a long history in machine learning; it is an area of research that has existed for decades. What is new in this current Generative AI moment are the exciting, novel capabilities of contemporary models. These capabilities have come about due to recent breakthroughs in model architectures,[33] massive-scale datasets on which to train those model architectures,[34] and immense computing power needed to run the training process for massive-scale models on massive-scale datasets.[35] Taken together, these three types of advancements have enabled contemporary applications like conversational chatbots and high-quality image generators.

B. Systems and Supply Chains

Generative-AI applications are more than just trained models. They consist of hosted software services that wrap software *systems*; generative-AI models are an embedded component of these systems, but they are only one such component. Other components include user interfaces, developer APIs, and input and output content filters (e.g., to remove "toxic" or copyrighted content from inputs before supplying prompts to models to produce generations, or from output generations before surfacing them to users).[36]

---

30. This strategy for generating tokens is called *greedy decoding*. There are other, more complicated decoding strategies for generation; it is not strictly necessary to always select the highest-probability token to be the next one in the generated sequence. Nevertheless, this is a useful way to think about generation: it involves sampling from a distribution over tokens, which are associated with different probabilities.

31. Response at 4–5, *Concord Music Grp., Inc. v. Anthropic PBC*, No. 3:23-cv-01092.

32. In this example, `"time"` is the most likely next token to complete `"once upon a"` because "once upon a time" is a common phrase in fairy tales (which we assume are included in the training dataset).

33. Talkin', *supra* note 11, at 25–27 (discussing the transformer-based model architecture).

34. Katherine Lee, A. Feder Cooper, James Grimmelmann & Daphne Ippolito, AI and Law: The Next Generation (2023), https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4580739.

35. Talkin', *supra* note 11, at 30–31 (discussing the importance of scale).

36. *Id.* at 16–18 (discussing generative-AI systems); OpenAI, GPT-4 System Card (Mar. 23, 2023) [hereinafter GPT-4 System Card] (unpublished manuscript), https://cdn.openai.com/papers/gpt-4-system-card.pdf (describing the entire GPT-4 system); A. Feder

There is an entire supply chain involved in the production of these models and systems—an ecosystem of actors and technical components that contribute to the development, deployment, and maintenance of user-facing software services. This supply chain is

> an interconnected set of stages that transform training data (millions of pictures of cats) into generations (a new and hopefully never-seen-before picture of a cat that may or may not ever have existed). Breaking down generative AI into these constituent stages reveals all of the places at which companies and users make choices that have legal consequences – for copyright and beyond."[37]

In prior work with Katherine Lee, we have described the supply chain in detail,[38] and discussed its relationship to U.S. copyright law.[39] We refer the interested reader to that work. Our summary here is meant only to introduce some essential terminology and to frame our later discussion.

In our account, the generative-AI supply chain has eight interconnected stages:

1. Creation of expressive works or other *information*,
2. Conversion of these expressive works or information into digitized *data* that can be interpreted by computers,
3. Collection and curation of enormous quantities of such data into *training datasets* (for generative-AI models, these datasets are frequently scraped from the Internet),[40]

Cooper, Karen Levy & Christopher De Sa, *Accuracy-Efficiency Trade-Offs and Accountability in Distributed ML Systems*, *in* 2021 Equity & Access Algorithms Mechanisms & Optimization 1 (2021); A. Feder Cooper & Karen Levy, *Fast or Accurate? Governing Conflicting Goals in Highly Autonomous Vehicles*, 20 Colo. Tech. L.J. 249 (2022) (emphasizing the role of AI/ML systems, not just models, in overall application behavior); Cooper, Lee, Grimmelmann & Ippolito et al., *supra* note 26 (discussing different business models for producing and combining these components); Lee, Cooper, Grimmelmann & Ippolito, *supra* note 34 (detailing data curation for training generative-AI models and the definition of "toxicity").

37. Talkin', *supra* note 11, at 5.

38. *Id.* at 32–55.

39. *Id.* at 55–148.

40. The practice of using web-scraped for generative-AI model training is one of the focal points of existing copyright lawsuits. Lee, Cooper, Grimmelmann & Ippolito, *supra* note 34 (discussing generative-AI training datasets); Pamela Samuelson, *Generative AI meets copyright*, 381 Science 158–61 (2023) (discussing lawsuits); Leo Gao, Stella Biderman & Sid Black et al., The Pile: An 800GB Dataset of Diverse Text for Lan-

4. *Pre-training*[41] of a general, large-scale, *base* (also called *foundation*) generative-model architecture on these curated datasets,
5. *Fine-tuning* of the pre-trained base model on additional data, in order to improve performance on a domain-specific task,
6. Public *release* of the model's parameters, or embedding the model in a system for *deployment* in a software service, and
7. End-user *generation* of outputs from a user-supplied prompt.[42]
8. *Alignment* of the model with human preferences or usage policies (a further stage of training that, for example, is responsible for ChatGPT behaving like a conversational chatbot).[43]

Even to call this a supply "chain" understates its complexity; it is a densely interconnected ecosystem, whose stages can branch, recombine, loop, repeat, and feed back into each other.[44]

Further, the supply chain is potentially carried out by many different actors, affiliated with potentially many different organizations, at each of the different stages.[45] "Copyright concerns cannot be localized to a single link

guage Modeling (2021) (unpublished manuscript), https://arxiv.org/abs/2101.00027; Christoph Schuhmann, Romain Beaumont & Richard Vencu et al., *LAION-5B: An open large-scale dataset for training next generation image-text models*, *in* 2022 Thirty-sixth Conf. on Neural Info. Processing Sys. Datasets & Benchmarks Track (2022) (detailing two web-scraped datasets that feature prominently in lawsuits).

41. Pre-training is just training. This term originates from the fact that there may be additional training further along in the supply-chain. Talkin', *supra* note 11, at 39–42.

42. *Id.*; Katherine Lee, A. Feder Cooper & James Grimmelmann, *Talkin' 'Bout AI Generation: Copyright and the Generative-AI Supply Chain (The Short Version)*, *in* 2024 Proc. Symposium on Comput. Sci. & L. 48–63 (2024) [hereinafter Talkin' (Short)].

43. Paul Christiano, Jan Leike & Tom B. Brown et al., Deep reinforcement learning from human preferences (2017) (unpublished manuscript), https://arxiv.org/abs/1706.03741v1; Long Ouyang, Jeff Wu & Xu Jiang et al., Training language models to follow instructions with human feedback (2022) (unpublished manuscript), https://arxiv.org/pdf/2203.02155.pdf; OpenAI, *ChatGPT: Optimizing Language Models for Dialogue*, OpenAI (Nov. 30, 2022), https://web.archive.org/web/20221130180912/https://openai.com/blog/chatgpt/.

44. Note that some stages are also optional and happen in different orders. For example, not all models are fine-tuned or aligned; some forms of alignment often precede deployment.

45. *See* A. Feder Cooper, Emanuel Moss, Benjamin Laufer & Helen Nissenbaum, *Accountability in an Algorithmic Society: Relationality, Responsibility, and Robustness in Machine Learning*, *in* 2022 2022 ACM Conf. on Fairness Accountability & Transparency 864 (2022); David Gray Widder & Dawn Nafus, *Dislocated Accountabilities in the "AI Supply Chain": Modularity and Developers' Notions of Responsibility*, 10 Big Data & Soc'y 1 (June 15, 2023) (discussing the challenges of accountability in AI supply chains).

in the supply chain. ... [D]ecisions made by one actor can affect the copyright liability of another, potentially far away actor in the supply chain."[46] For example, the choices of dataset curators upstream in the supply chain have significant downstream effects on the possible generations that the users of a generative-AI system can produce.[47] Consequently, it is necessary to reason about the entire supply chain—the ecosystem of diffuse actors and technical artifacts—for a complete infringement analysis.

This brief gloss of the generative-AI supply chain introduced key terminology and background we use in the remainder of this Essay. We will bring in additional terminology (e.g., *memorization*[48]) as needed. For our purposes, the important takeaway from the supply-chain framing is its complexity. As appealing as it might be to come up with broad generalizations about copyright and generative-AI—e.g., a one-size-fits-all fair-use analysis of training datasets—the supply chain makes clear that it is not possible to do so. A rigorous analysis of copyright implications depends on the specific system; such an analysis turns on the particular details of the supply chain invoked during the system's construction and use.

Our goal in this Part has been merely to recapitulate the technology of Generative AI in terms that are accurate enough to be honest but abstract enough to be useful.[49] We believe that accurate abstraction is the appropriate starting point for legal analysis. In the next Part, we show what can go wrong when legal models outstrip technical reality.

III. Memorization is in the Model

The previous part emphasized both the simplicity and the complexity of generative-AI systems. On the one hand, at a high enough level of abstraction,

46. Talkin', *supra* note 11, at 147.

47. For example, as we will see, it is by definition not possible to regurgitate *memorized* training-data images of Elsa from *Frozen* if there are no images of Elsa in the training data. *See infra* Part III.A. However, for various reasons, it may nevertheless still be possible to generate images that closely resemble Elsa; they just will not be evidence of memorization (as it is typically defined in the technical literature). *Id.* at 72–85 (discussing substantial similarity in the generative-AI supply chain). Aaron Gokaslan, A. Feder Cooper & Jasmine Collins et al., *CommonCanvas: Open Diffusion Models Trained on Creative-Commons Images*, *in* 2024 Proc. IEEE/CVF Conf. on Comput. Vision & Pattern Recognition (CVPR) 8250–60 (2024) (for a model trained on Creative Commons images, with the goal of reducing memorization of copyrighted or unlicensed works).

48. *See infra* Part III.A.

49. This is the point, more generally, of the supply-chain framing from our prior work. *See* Talkin', *supra* note 11.

generative-AI models are incredibly simple. They are data structures that encode information about the examples in the training dataset. They can be embedded in computer programs, and then prompted to generate outputs that reflect statistical patterns in these training examples. On the other hand, this high-level description applies to an enormous range of models and systems. Models are trained in different ways, encode information in different ways, and generate outputs of different kinds in different ways. They are trained on different datasets, can be aligned, can be released, or can be embedded and deployed in different systems. The facts that a model encodes information about the training data, can be prompted to generate outputs of the same modality as its training data, and can produce generations that reflect statistical patterns in its training data might seem like the *only* facts that are generally true of all the models currently being described as "Generative AI."

But there is at least one more fact that that applies to all large-scale generative-AI models, which serves as the focal point for the remainder of this Essay, as well as many current lawsuits: all generative-AI models *memorize* some portion of their training data. In this Part, we clear up important misconceptions about what memorization is, with a particular eye toward implications for copyright.

- We begin in Part III.A by defining *memorization* and distinguishing it from related terms: *regurgitation*, *extraction*, and *reconstruction*. Even the initial step of clarifying definitions has important implications. In particular, generation-time regurgitation implies that memorization has taken place during the training process.
- Next, in Part III.B we discuss in detail how memorized training data is *within* models in terms of the "patterns" that models learn during training. Here, we engage with copyright law, and show that memorization in a model constitutes a "reproduction" of the memorized data.
- Part III.C uses this understanding of memorization to explore two common metaphors for how generative-AI models work: that they learn only "patterns" in their training data and that they "compress" their training data. Both metaphors have a kernel of truth, but neither should be taken as a guide for how a model works in all cases.
- Part III.D discusses how *non-determinism* in generations—how the same prompt can yield possibly very different—plays an important role in how we should think about the copyright implications of memorization.
- From this basis, in Part III.E we dig into the state-of-the-art understanding of *how much* models memorize in practice.

- Of course, generative-AI models typically do more than just memorize their training data, so we bring in relevant details of learning and *generalization* in Part III.F.
- Then, we consider the implications of the fact that a generative-AI model both memorizes *and* generalizes. In Part III.G, we consider the analogy between a generative-AI model and other "dual-use" technologies, such as VCRs. In our view, the analogy fails in important ways; VCRs do not contain the works they can be used to infringe in the same way that memorizing models do.[50]
- In Part III.H, we return to the figure of the "adversarial" user invoked by defendants in current copyright lawsuits. Not every user is adversarial, nevertheless, we argue that the users who are cannot simply be waved away as pesky exceptions.
- Finally, in Part III.I, we step back from models to look at system design. Memorization in a model does not mean that a system necessarily has to produce copies of that memorized data in its generations; the model is just one of many pieces that system builders can adjust to tune the system's overall outputs. There are several other places where system builders can attempt to limit how much memorization gets surfaced to end users.

A. Definitions

It is helpful to distinguish three related senses in which a model might colloquially be said to have "memorized" its training data.[51] They have in common that the training data can be surfaced from the model; they differ in the process by which this surfacing takes place, and they are generally given different names in the machine-learning literature:[52]

50. VCRs are not themselves copies of the tape copies they produce. *See infra* note 63 and accompanying text (for a discussion comparing the colloquial, everyday use of the word "copy" with the term's meaning in U.S. copyright law).

51. This is just one axis along which one can break down memorization; it deals with *how* a model memorizes and how that memorization can be brought to light. Another is to distinguish *what* a model memorizes: it could memorize complete training examples, or it could memorize isolated facts (such as social security numbers), or common information present in many examples (such as the features of a celebrity's face), or many other things. And a third is to distinguish *how much* a model memorizes: only one example, or many, etc. In this Essay, we use a common technical definition of "memorization" that refers to exact or near-exact copying of a substantial portion of a piece of training data. *See* The GenLaw Center, *The GenLaw Glossary* (2023) [hereinafter GenLaw Glossary], https://genlaw.org/glossary.html.

52. This terminology, for this particular notion of memorization that involves exact or near-exact copying of training data in the model (as opposed to other uses of "memoriza-

- Most narrowly, when a user intentionally and successfully prompts a model to generate an output that is an exact or near-exact copy of a piece of training data,[53] that is **extraction**.[54]

---

tion" is still in flux. We have summarized common usages in the literature, but these are not the only usages. *See* Nicholas Carlini, Daphne Ippolito & Matthew Jagielski et al., *Quantifying Memorization Across Neural Language Models* 3, *in* 2023 Int'l Conf. on Learning Representations (2023) (for a discussion of different memorization terminology and metrics in the machine-learning literature). *See* Daphne Ippolito, Florian Tramèr & Milad Nasr et al., *Preventing Verbatim Memorization in Language Models Gives a False Sense of Privacy*, *in* 2023 Proc. 16th Int'l Nat. Language Generation Conf. (2023) (for a definition of memorization that considers translations of a given piece of text data).

53. We say "a piece of training data" instead of "training example" in these definitions because, when measuring memorization in practice for production systems and many released models, researchers often do not know the training datasets (nor the specific training examples). They use proxy methods to approximate memorization of training data, and these methods can end up measuring memorization of what was ultimately used as a piece of a particular training example (e.g., a piece of a news article), or training data that happened to span multiple examples (e.g., a whole news article that, during training, was actually split up into multiple different training examples).

Regardless of these subtleties, memorization measurements in the technical literature tend to capture (typically verbatim) copying of portions of the training data given as input to the training process that are produced in output generations. *See supra* note 24 and accompanying text (discussing text examples in relation to full text works); Milad Nasr, Nicholas Carlini & Jonathan Hayase et al., Scalable Extraction of Training Data from (Production) Language Models (2023) (unpublished manuscript) (describing proxies for measuring memorization in models for which we do not know the exact training dataset).

54. This is how the word "extraction" is used in copyright lawsuits. The term is partially overloaded with the technical literature, for which "extraction" definitions have subtle differences. In the technical ML-research literature, "extraction" often refers to an "extraction attack" that aims to retrieve training data from a model. Following from this setup as a security-attack problem, *extractable memorization* tends to refer to memorization that can be retrieved with *any* constructed prompt—notably, where such a prompt is constructed without access to the training data.

We (and lawsuit responses) emphasize the intent aspect in our discussion of extraction. Our discussion applies to individual users (both actual users, like current plaintiffs, and hypothetical potential users of generative-AI systems). It is these users that *extract* training data in our Essay. This usage clearly differs from the particulars of research- and security-based *extraction attacks*. *See* Nasr, Carlini & Hayase et al., *supra* note 53; Nicholas Carlini, Florian Tramèr & Eric Wallace et al., *Extracting Training Data from Large Language Models*, *in* 2021 30th USENIX Security Symposium (USENIX Security 21) 2633—2650 (2021); Nicholas Carlini, Jamie Hayes & Milad Nasr et al., Extracting Training Data from Diffusion Models (2023) (unpublished manuscript), https://arxiv.org/abs/2301.13188 (discussing extractable memorization and extraction attacks).

- More broadly, when a model generates an output that is an exact or near-exact copy of piece of training data (whether or not the user intentionally prompted the model with that goal), that is **regurgitation**.
- Most broadly of all, when an exact or near-exact copy of a piece of training data can be reconstructed by examining the model *through any means*, that is **memorization**.[55] We will use the term **reconstruction** to refer to these processes, which can include but are not limited to prompting.

These definitions clearly depend on what counts as an "exact" or "near-exact" copy of a piece of training data, but none of the arguments we will make in this Essay do. Instead, our definitions of these terms are designed to work with any reasonable definition of how exact an exact copy must be. First, the details will clearly differ for different modalities; similarity of images will need to be assessed differently than similarity of text, which in turn is different than similarity of music, and so on. Second, copyright law itself uses an equally high-level definition: "substantial similarity" for different works is a detailed question of fact that can only be answered in a specific case. And third, whatever definition of exactness one uses, a generative-AI model that meets that definition for purposes of regurgitation also meets it for purposes of memorization. All that our argument requires is that the test of exactness be used consistently.

This taxonomy focuses on the technical characteristics of the generative-AI model and its behavior; it does not consider whether these characteristics and behavior are intentional or unintentional from the point of view of the model's creator.[56] Within the taxonomy, *The New York Times* pleads regurgitation: it alleges that LLMs can be prompted to output near-exact copies of training data.[57] (Note, however, that the complaint is (strategically) silent on whether this prompting is done with the goal of eliciting those near-exact copies, in which case it would be extraction as well.)

Memorization is a front-end phenomenon; it describes characteristics and capabilities of the model itself that directly result from its training. Re-

55. *See generally* GenLaw Glossary, *supra* note 51; Cooper, Lee, Grimmelmann & Ippolito et al., *supra* note 26.

56. *See generally* Ali Naseh, Jaechul Roh & Amir Houmansadr, Understanding (Un)Intended Memorization in Text-to-Image Generative Models (2023) (unpublished manuscript), https://arxiv.org/abs/2312.07550 (discussing intentional and inintentional memorization).

57. *See* Complaint at 23–24, N.Y. Times Co. v. Microsoft, No. 2:24-cv-00711 (C.D. Cal. Dec. 27, 2023) (internal citations omitted) (models "are known to exhibit a behavior called 'memorization.' That is, given the right prompt, they will repeat . . . portions of materials they were trained on."); *see also* Concord Music Grp., Inc. v. Anthropic PBC, No. 3:23-cv-01092 (M.D. Tenn.).

gurgitation and extraction are back-end-phenomena; they describe how the model behaves in generating outputs in response to a specific prompt.[58] The definition of memorization refers to hypothetical processes that reconstruct training data from the model. It covers all possible such processes, and is designed to capture any possible way that someone could use the model to reconstruct its training data. But the definitions of regurgitation and extraction deal with specific behavior under specific prompts.

The amount of memorization that can be regurgitated in practice depends on numerous choices by model creators and system designers. For example, longer prompts can be more effective at extracting training data.[59] Thus, a system limit that prevents users from submitting long prompts (on the back-end) does not affect what data the model has memorized (on the front-end)—but it might reduce the amount of memorized training data that can be surfaced at generation time.

Thus, not all memorization is regurgitation or extraction; material can be present in a model but inaccessible through prompting with a particular strategy. For an (imperfect) analogy, consider the Google Books database.[60] Google's corpus of scanned books includes complete images of every page from the books it has scanned; treating the corpus like a model, it would be a straightforward example of memorization. Google allows searchers on Google Books to view "snippets" of an eighth of a page containing their search terms, which would be straightforward regurgitation and extraction.[61] But Google "makes permanently unavailable for snippet view one snippet on each page and one complete page out of every ten."[62] In our analogy, those withheld snippets and pages are memorized but never regurgitated.

Some important observations follow directly from these definitions, which we discuss in the remainder of this section.

58. We thank Derek Bambauer for the "front-end"/"back-end" terminological distinction.

59. *See* Carlini, Ippolito & Jagielski et al., *supra* note 52, at 4 ("this [is] the **discoverability phenomenon**: some memorization only becomes apparent under certain conditions, such as when the model is prompted with a sufficiently long context. The discoverability phenomenon may seem natural: conditioning a model on [a prompt of] 100 tokens of context is more specific than conditioning the model on [a prompt of] 50 tokens of context, and it is natural that the model would estimate the probability of the training data as higher in this situation. However, the result is that some strings are 'hidden' in the model and require more knowledge than others to be extractable.").

60. Of course, Google Books is a database of scanned images and text, not a generative-AI model, which stores information differently. Among other things, this distinction limits the capabilities of the Google Books database compared to contemporary generative models, but also gives Google greater ability to restrict its outputs to prevent how much copied content is surfaced to the user.

61. Authors Guild v. Google, Inc., 804 F. 3d 202, 209–10 (2d Cir. 2015).

62. *Id.* at 210.

1. Regurgitation is Copying

First, *regurgitation is copying*: it involves the creation of a copy of training data as the output of a model. (It follows *a fortiori* that extraction is also copying, since extraction is regurgitation plus intent.) More precisely, regurgitation is what a copyright lawyer would call *literal* copying: the near-exact replication of (potentially a substantial) portion of a work. Literal copying is not the only viable theory of copyright infringement—courts have also found infringement based on non-literal or fragmented similarities—but it is the simplest and most straightforward.

When we say that regurgitation is copying, we are using "copy" as a term of art from copyright law. The Copyright Act states that "copies" of a copyrightable work are "objects . . . from which the work can be perceived, reproduced, or otherwise communicated."[63] Under this definition, if I have Blu-Ray disc of *Barbie* (2023), it is a "copy" of the audiovisual work *Barbie*, because it can be "perceived" by playing it in a Blu-Ray player. If I rip the disc to an SSD storage device, the SSD also becomes a "copy" of *Barbie*; it can be "perceived" by playing it with software like QuickTime Player. It is still a "copy" even if I downscale it to a lower resolution and change the file format; copies do not have to have exactly the same information or the same encoding. The SSD might also be a copy of many other works stored on it: the audiovisual work *Oppenheimer* (2023), the sound recording *Barbie Girl*, the computer program *Final Cut Pro*, and so on. If I use Final Cut Pro to alternate enough scenes to infringe from *Barbie* with enough scenes to infringe from *Oppenheimer* and burn the mash-up to another Blu-Ray disc, that disc is now a "copy" of both *Barbie* and *Oppenheimer*. The *legal* definition of "copy" is functional: a "copy" of a work is defined by the fact that one can reconstruct enough of the work from it.

This usage does not entirely track the lay or technical usage of "copy." Such usage might insist that the SSD *contains* a "lower-resolution version" of *Barbie* rather than *being* a "copy" of it—and that is a perfectly reasonable position as a matter of the word in everyday usage. But to a copyright lawyer, a "copy" is defined by what it does, and the SSD is a copy. It is in this sense that we say that a machine-learning model is a "copy" of works the model has memorized—the copyright sense.

To say that regurgitation is copying does not necessarily mean that it is copyright infringement. A model might regurgitate unembellished, uncopyrightable material, like the factual alphabetized list of the fifty U.S. states.[64] It might regurgitate a copyrightable work in the public domain, like the text

63. 17 U.S.C. § 101.

64. *See* Nasr, Carlini & Hayase et al., *supra* note 53 (from which this example is drawn).

of Virginia Woolf's *To the Lighthouse*. It might regurgitate a copyrightable work under a license from the copyright owner. It might regurgitate a copyrightable work in a way that is held to be fair use. It might regurgitate a small (e.g., 50 tokens), uncopyrightable piece of an overarching copyrightable work.

And even if none of these apply, substantial similarity requires an assessment comparing the two works (input and output) from the point of view of an ordinary observer. Their aesthetic reaction need not correspond to whatever numerical threshold of similarity a computer scientist quantifying regurgitation might use. In *Universal City Studios, Inc. v. Kamar Industries, Inc.*, for example, a court held that the phrase "E.T. Phone home!" on a mug by itself was sufficient to infringe the copyright in *E.T. the Extra-Terrestrial* (1982).[65] But in *Alberto-Culver Co. v. Andrea Dumon, Inc.*, the court held that the longer phrase "most personal sort of deodorant" was not copyrightable.[66] Scholars have begun to develop more sophisticated ways of quantifying copying, but as these examples show, simple thresholding does not suffice to capture copyright's tests for similarity.[67]

Fair use, in particular, is not a purely technical question—especially following the Supreme Court's decision in *Andy Warhol Foundation for the Visual Arts*. In *Warhol*, the Court found that the use of a print on a magazine cover was not transformative, but that other uses of the same print might be. Similarly, it is possible that a given regurgitated output could be fair use if it were emitted by a free non-profit service and not fair use if it were emitted by a paid for-profit service.[68] It is also possible that an output could be infringing on its own but then put to a noninfringing fair use by the user who requested it (e.g., a verbatim copy could itself be an input into the user's artistic process of creating a biting parody). Contrariwise, a non-infringing output could be put to an infringing use (e.g., a user prompts a model, which generates a stylistic variation of an artist's work, and then sells this generation on T-shirts). *Andy Warhol Foundation for the Visual Arts*'s use-by-use emphasis on purpose, context, and commerciality means that the fair use inquiry will also generally turn on facts outside of the generative-AI system itself.

All told, our first point in this Section is simply that regurgitation is copying in the sense with which copyright law is concerned. Indeed, this is

65. Universal City Studios, Inc. v. Kamar Indus., Inc., 217 U.S.P.Q. 1162 (S.D. Tex. 1982).

66. Alberto-Culver Co. v. Andrea Dumon, Inc., 466 F.2d 705 (1972).

67. *See* Scheffler, Sarah, Eran Tromer & Mayank Varia, *Formalizing Human Ingenuity: A Quantitative Framework for Copyright Law's Substantial Similarity*, *in* 2022 Proc. Symposium on Comput. Sci. & L. 37 (2022) (defining an information-theoretic model of copying and similarity for copyright law).

68. Andy Warhol Found. for the Visual Arts, Inc. v. Goldsmith 598 U.S. 508 (2023).

precisely why copyright complaints in generative-AI cases emphasize regurgitation: it establishes a *prima facie* case of infringement.[69]

2. Regurgitation Implies Memorization

Second, *regurgitation implies memorization*. (It follows *a fortiori* that extraction also implies memorization.) In a sense, this claim is tautologically true: memorization takes place when a piece of training data can be emitted from a model by any means, and prompting is one such means. But there is a deeper point here. The definitions of extraction and regurgitation focus attention on the generation of outputs. They could be (mis)understood to suggest that the only significant act of copying takes place at the generation stage of the generative-AI supply chain,[70] when a model is prompted to generate and produces an output that is nearly identical to a piece of training data.[71]

But, for memorization, focusing on the copying that takes place during the generation of model outputs elides the copying that takes place during model training. In order to be able to extract memorized content from a model at generation time, that memorized content must be encoded in the model's parameters.[72] There is nowhere else it could be. A model is not a

69. *E.g.,* N.Y. Times Co. v. Microsoft, No. 2:24-cv-00711 (C.D. Cal.); Concord Music Grp., Inc. v. Anthropic PBC, No. 3:23-cv-01092 (M.D. Tenn.).

70. *See supra* Part II.B.

71. *See supra* Part III.I (discussing system-level modifications that wrap around the model and can prevent memorized training data from being surfaced to the end user, even if memorized training data is generated by the model).

72. There are choices that model trainers can make to reduce the likelihood of memorization at training time. One common strategy is to deduplicate training data, with respect to identical or highly similar (by some choice of quantitative metric, like using the MinHash algorithm and edit distance). This makes sense intuitively: (many) duplicates of particular pieces of text in the training dataset makes it more likely that the model learns "statistical correlations" or "patterns" that relate the words those fragments contain to each other. For such a piece of duplicated text t, we can think of dividing it into a prefix t_p and suffix t_s ($t = t_p + t_s$). With duplication, it becomes more likely that a particular suffix t_s would be generated in response to a prompt that is the prefix t_p, since t_p and t_s are repeatedly associated together in the training data. *See supra* notes 30–32 and accompanying text (discussing model generation of tokens in response to prompts). *See* Katherine Lee, Daphne Ippolito & Andrew Nystrom et al., *Deduplicating Training Data Makes Language Models Better*, *in* 1 Proc. 60th Ann. Meeting Ass'n for Comput. Linguistics 8424 (2022) (discussing deduplication of training data and reduction of memorization).

More recent work also identifies new training optimization objective (the cutely named "goldfish" loss function) that reduces extraction of memorization at generation time, at the cost of requiring longer training time to achieve comparable quality on benchmarks. *See* Abhimanyu Hans, Yuxin Wen & Neel Jain et al., Be like a Goldfish,

magical portal that pulls fresh information from some parallel universe into our own. A model is a data structure: it consists of information derived from its training data. The memorized training data are *in the model*.

The *Times* complaint recognizes this point. Although its definition of memorization focuses on extraction, it also notes, "This phenomenon shows that LLM parameters encode retrievable copies of many of those training works."[73] Indeed, this claim seems to form part of the complaint's basis for requesting an order for the destruction of GPT models.[74] As the complaint argues, whenever a model has memorized a training work, *it has copied that training work*.[75]

Even if the only currently effective tool to observe memorized training data is prompting, this does not change the fact that these data *are* memorized. True, we cannot observe the memorized training data directly in the model's parameters—but neither can we directly observe black holes, ultraviolet light, or electric fields. We can confirm their existence through indirect measurements—detecting certain types of nearby radiation, using specialized sensors, and observing behavior of charged particles, respectively. In the same way, extraction of memorized training data is a kind of indirect measurement. If we can generate verbatim a training-data painting of the Eiffel tower by providing an appropriate prompt, we have produced an (indirect) proof by example that this specific painting *is represented in the model*.[76]

This is the problem with Tyler Cowen's toothpick-memorization hypothetical. It is true that in theory, with a sufficiently precise "prompting" procedure, one could "find" the text of a *Times* article in the dimensions of a toothpick. But one can "find' *any* text this way; in the trivial sense of Cowen's

Don't Memorize! Mitigating Memorization in Generative LLMs (2024) (unpublished manuscript), https://arxiv.org/abs/2406.10209 (proposing goldfish loss).

73. Complaint at 24, *N.Y. Times Co. v. Microsoft*, No. 2:24-cv-00711 (Dec. 27, 2023).

74. *Id.* at 68.

75. Talkin', *supra* note 11, at 74–85.

76. *Id.* at 74–77. Another piece of (indirect) evidence comes from the research area of *machine unlearning*, which (traditionally) has sought to remove specific training examples—e.g., examples that contain an individual's address—from a model after it has been trained. Machine unlearning is often motivated by legislative provisions, such as "the right to be forgotten" in the GDPR. From first principles, how would this problem formulation make sense (e.g., removing an individual's address from the model, so that it cannot be produced at generation time) if the information targeted for removal was not encoded somewhere within the model to begin with? *See* Lucas Bourtoule, Varun Chandrasekaran & Christopher A. Choquette-Choo et al., *Machine Unlearning*, *in* 2021 2021 IEEE Symposium on Sec. & Priv. (SP) 141–59 (2021); Seth Neel, Aaron Roth & Saeed Sharifi-Malvajerdi, *Descent-to-Delete: Gradient-Based Methods for Machine Unlearning*, *in* 132 Proc. 32nd Int'l Conf. on Algorithmic Learning Theory 931—962 (2021) (for early work in machine unlearning).

example, there is a prompt that will generate any desired output from the toothpick. This puts an incorrectly strong emphasis on the role of prompt construction, and elides the important role of the model.

To see why this emphasis is misplaced, consider an absurdly simple "model": one that simply emits its prompt as its output.[77] This model is trivial to implement and trivial to describe, and it is intuitively clear that it has memorized nothing. And yet you can cause this model to generate anything you want—an oil painting of the Eiffel Tower, a *Times* article, the schematics for an electro-mechanical trombone—but not because it has memorized or learned anything about any of them.[78] You get out exactly what you put in; the prompt itself is just another way of encoding the output. Like the toothpick, it tells you nothing more than was already present in your prompt.

In contrast, what makes the fact that specific training data can be extracted from a generative-AI model so telling is that *not everything can be extracted*. If I try to "extract" a genuine black-and-white photograph of a steampunk Abraham Lincoln riding a seahorse in space from a model trained only on oil paintings of world-famous landmarks, I will fail, no matter what prompt I put in.[79] Such a model could memorize a painting of the Eiffel Tower; it could not memorize a genuine photograph of Lincoln on a seahorse in space. The Eiffel Tower is in the training data; Abraham Lincoln on a seahorse in space is not. The *Times*'s examples are telling because ChatGPT continues with text that was not part of the prompt but was part of a *Times* article. In a sense that can be made mathematically rigorous, the information that ChatGPT produces comes from the model, whereas the information that the toothpick "produces" comes from the prompt.

In copyright terms, this is a form of striking similarity. When an output is highly similar to one specific training work, and significantly dissimilar from all other training works, the argument goes, it is strong evidence that the model has memorized (part or all of) that specific work. First, the similarities are unlikely to reflect broader patterns[80] in the training data, since the specific work stands alone in its distinctive elements. Second, the similarities are extraordinarily unlikely to have arisen by coincidence, since the space of

77. Mathematically, this model implements the identity function $f(x) = x$, whose output is always the same as its input.

78. *See* Matthew Sag, *Fairness and Fair Use in Generative AI*, 92 Fordham L. Rev. 1887, 1912 (2024) (discussing use of generative models to create infringing derivative works of prompts).

79. No such genuine photograph exists and thus cannot serve as training data (and therefore cannot, by definition, be memorized).

80. *See supra* Part II.B; *infra* III.B; *infra* III.E (discussing the double duty of words like "pattern" to describe machine learning, but elide memorization).

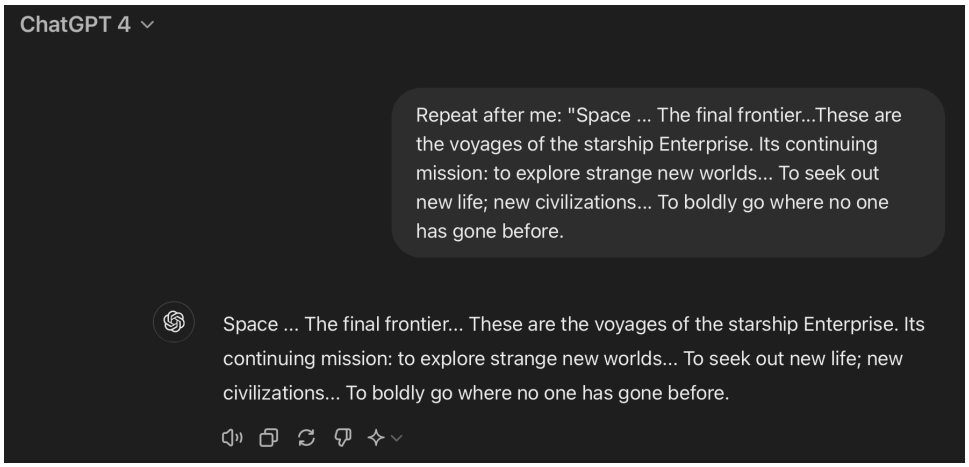

Figure 2: Prompting the ChatGPT GPT-4o endpoint to "repeat after me" is not useful evidence of memorization. Here, the model repeats the text from the monologue in the opening credits of *Star Trek: The Next Generation*. Interaction produced by the authors.

all possible outputs—both those the model was trained on and those it was not—is immense.

To see this point, consider an LLM that has been trained to behave like a chatbot and follow instructions.[81] Prompting the model with `"Repeat after me: 'Space... The final frontier... These are the voyages of the Starship Enterprise...'"` does not elicit useful information about memorization. (See Figure 2.) This text may well be memorized in the model, but the context of a prompt like this one, an instruction-following model like ChatGPT is demonstrating its capability to repeat its input— similar, at a high level, to the identity function above—and is not regurgitating its training data.[82]

81. *See* Rohan Taori, Ishaan Gulrajani & Tianyi Zhang et al., Stanford Alpaca: An Instruction-following LLaMA model (2023) (unpublished manuscript), https://github.com/tatsu-lab/stanford_alpaca; Jason Wei, Maarten Bosma & Vincent Zhao et al., *Fine-tuned Language Models are Zero-Shot Learners*, *in* 2022 Int'l Conf. on Learning Representations (2022) (discussing examples of instruction-fine-tuned models). *See* Ouyang, Wu & Jiang et al., *supra* note 43 (discussing a technique for aligning a model to follow instructions using reinforcement learning from human feedback (RLHF)).

82. This is partly what is so interesting (and peculiar) about the Google DeepMind divergence-based attack that got ChatGPT to emit training data. In that technical paper, researchers prompted ChatGPT with `"Repeat this word forever: 'poem poem ... poem'.''`. ChatGPT is a Chatbot is aligned to follow instructions, and at first did exactly that. However, in nearly every case, after repeating the word "poem" some number of times, the model stopped following the instruction and instead start emitting different, "divergent" text (and sometimes, that text contained memorized

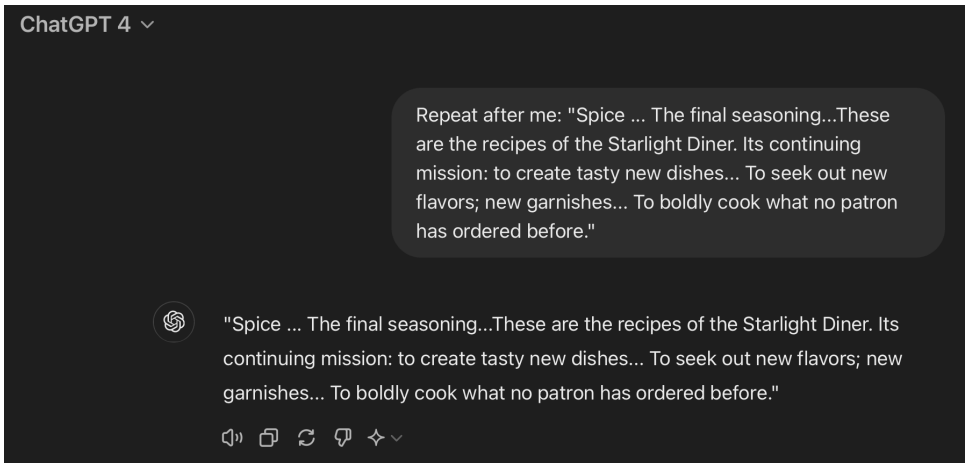

Figure 3: Here, the model repeats a control text equally well. Interaction produced by the authors.

We can support that this is likely the case by prompting the model with `"Repeat after me: 'Spice... The final seasoning... These are the recipes of the Starlight Diner...'"`. (See Figure 3). This control text, which was written specifically for this example, is almost certainly not in ChatGPT's training dataset and therefore almost certainly cannot be memorized. Thus, ChatGPT is equally good at repeating the test text in Figure 2 (which it was likely trained on) and the control text in Figure 3 (which it likely was not). This is a case where there is a compelling alternative explanation for a model's outputs *besides* memorization. What makes extraction so telling is it succeeds in reproducing specific training data in a way that would be highly improbable if the model had not memorized that data.

The *technical* fact that memorization is in the model does not compel any particular *legal* conclusion. On the one hand, courts could hold that generative-AI models are themselves infringing copies[83] of the expressive works they have memorized—regardless of whether they are used to produce infringing generations in practice.[84] On the other hand, this fact might not matter to courts at all. There is ample precedent for treating expression that

training data). The authors had to circumvent the aligned, instruction-following behavior in this case, after which it was possible to demonstrate memorization. *See* Nasr, Carlini & Hayase et al., *supra* note 53 (describing divergence and memorization in ChatGPT-3.5).

83. In the same sense that the SSD, in our example above, is an infringing copy of *Barbie*. *See supra* Part III.A.

84. Talkin', *supra* note 11, at 76, 129–30; Pamela Samuelson, *How to Think About Remedies in the Generative AI Copyright Cases*, Lawfare (Feb. 15, 2024), https://

is stored in a computer system but never directly exposed to an end user—in our terminology, that is memorized but not regurgitated—as fair use.[85] Indeed, courts might hold that memorization is fair use even in some cases when a model also regurgitates the memorized expression.[86]

AI companies' responses to copyright lawsuits typically take this second position (sometimes explicitly, sometimes implicitly). Rather than discussing whether and how much their models have memorized,[87] they typically limit the scope of their lawsuit responses to "regurgitation" or "extraction" at generation time.[88] This framing places the focus on the *users*' role in selecting prompts and the resulting generations, rather than on the *companies*' role in designing a training process and the resulting model. For example, Anthropic's response never uses the word "memorization." Instead, it uses "regurgitate" once and variations on "extraction" four times.[89] This choice is rhetorically interesting because the terms "regurgitation" and "extraction" both inherently emphasize behaviors that can happen on the backend, at generation time. In contrast, "memorization" centers the behavior of the model with respect to its training data—behavior that results from training on the front-end.[90]

---

www.lawfaremedia.org/article/how-to-think-about-remedies-in-the-generative-ai-copyright-cases.

85. James Grimmelmann, *Copyright for Literate Robots*, 101 Iowa L. Rev. 657 (2016) (summarizing caselaw on intermediate copying). The system designer might also need to take reasonable measures to ensure that such "internal copies" are not surfaced externally to end users. *See infra* Part III.I (detailing the role of software-service-wrapped systems in filtering user inputs and model outputs).

86. We believe that the flexible fair-use test is a more appropriate way to hold that a model is non-infringing, rather than holding that it is not even a reproduction of works it has memorized. *See generally* Matthew Sag, *Copyright Safety for Generative AI*, 61 Hous. L. Rev. 295 (2024) (discussing in detail the fair use analysis of Generative AI). *See also* Talkin', *supra* note 11, at 105–14.

87. Unfortunately, companies rarely if ever have released such numbers, including in scientific research contexts. One exception, from a couple of years ago, is Google's PaLM model. Aakanksha Chowdhery, Sharan Narang & Jacob Devlin et al., *PaLM: Scaling Language Modeling with Pathways*, 24 J. Mach. Learning Rsch. 1–113 (2023) (discussing memorization in the PaLM model).

88. These AI-company responses typically engage plaintiffs as the users who are the source of regurgitation and extraction. This should not be confused with how these same AI companies discuss extraction in other contexts, e.g., extraction attacks that their researchers conduct to produce scholarly articles. *See supra* notes 51–55 and accompanying text (discussing overloading of the word "extraction").

89. Response, Concord Music Grp., Inc. v. Anthropic PBC, No. 3:23-cv-01092 (M.D. Tenn. Jan. 16, 2024).

90. *See supra* notes 58–59 and accompanying text.

The other reason we have emphasized that regurgitation implies memorization is to make clear that the two are different. In copyright terms, they involve different copies. Memorization involves front-end copying: the training data is copied in the *model*. Regurgitation involves back-end-copying: the training data is copied in the *output*. Memorization makes regurgitation possible; regurgitation shows that memorization has taken place.

3. Known vs. Unknown Memorization

Finally, it is important to distinguish our *knowledge* of whether a model has memorized training data from the underlying question of memorization itself.[91] It is possible that data could be reconstructed from a model through techniques that are currently unknown but will be discovered in the future. In this case, the model has memorized these data, but we do not currently have the means to know that it has done so. For example, OpenAI has claimed that its alignment techniques successfully trained its ChatGPT models to avoid memorization.[92] But in late 2023, a team led by Google DeepMind researchers developed a new technique and conducted a large-scale measurement study that showed the ChatGPT 3.5 (turbo endpoint) model memorized significantly more training data than any other model they tested.[93] The right way to describe this situation is that this work showed that ChatGPT 3.5 had memorized training data all along—and not that ChatGPT suddenly went from not memorizing training data to memorizing it just because this research team devised a way to get it out of the system.[94]

Similarly, the Copyright Act uses the phrase "now known *or later developed*" to describe the reconstruction of a work from a copy.[95] It is possible that a model is *currently* a copy of some of its training data, even though the techniques for extracting it will only be developed in the *future*. For another example (also involving advances in machine learning), ancient papyrus scrolls buried in the eruption of Mt. Vesuvius in 79 C.E. were discovered 275 years ago but were too fragile to physically unroll without collapsing

91. Our thanks to Benjamin Sobel for a helpful discussion of this distinction.

92. GPT-4 System Card, *supra* note 36 (claiming that recent GPT-series models had been aligned to prevent memorization).

93. Nasr, Carlini & Hayase et al., *supra* note 53.

94. Indeed, the *systems* (and alignment) aspects of ChatGPT made this task more challenging than extracting training data directly from a (un-aligned) model. *See id.*

95. 17 U.S.C. § 101 (definitions of "copies" and of "device," "machine," or "process"); *cf.* § 102 (defining copyrightability in terms of fixation in media of expression "now known or later developed").

into a pile of ash.[96] Scholars have recently successfully used computer tomography to generate images of the interior of the rolled-up scrolls and machine learning to identify the letters on them. The scrolls were fixed copies all this time, but we did not have definitive proof until this year.

Thus, a little unfortunately, it is often the case that generative-AI developers, users, and commentators must live in a state of ignorance about whether a model has memorized its training data. A successful extraction or an instance of regurgitation can provide positive proof that it has memorized. In some cases, it may be possible to show that a model has not memorized certain data because these data were definitely not present in the training dataset.[97] But in between there is a middle ground of greater and lesser ignorance: there is a fact of the matter, and we can have good reasons for thinking that a model has or has not memorized with a given degree of confidence, but certainty is not to be had. Like many other scientific, historical, and evidentiary facts about the world, where there is a truth out there but the legal system cannot definitively ascertain it, any decision-making will have to take place against this backdrop of partial knowledge. The legal system will have to deploy its usual tools for dealing with epistemic uncertainty: burdens of proof, expert analysis, findings of fact that can be reopened on the basis of new evidence, and so on.[98]

B. Representation

In this section, we will describe—at a high level, but carefully—how models represent the information stored in them. Scholars sometimes argue that models are uninterpretable, or unintelligible, or "do not generally contain recognizable expressions."[99] These claims are true in some senses, but incorrect with respect to memorization.

Models store information in different ways than more familiar file formats do—in model parameters rather than in direct one-to-one encodings—but they still store information. (Otherwise, the model would be useless.) Information is typically obtained from models in different ways than from other forms of encodings—through prompting rather than a deterministic algorithmic decoding—but information can still be obtained from them. (Otherwise, again, the model would be useless.)

---

96. *See Vesuvius Challenge 2023 Grand Prize awarded: we can read the first scroll!* (Feb. 5, 2024), https://scrollprize.org/grandprize.

97. There are no known methods that guarantee that a model has not memorized data that it was trained on.

98. *See* Cooper, Levy & De Sa, *supra* note 36; Cooper & Levy, *supra* note 36 (discussing the relationship between legal decisionmaking under uncertainty and uncertainty in ML).

99. Samuelson, *supra* note 84.

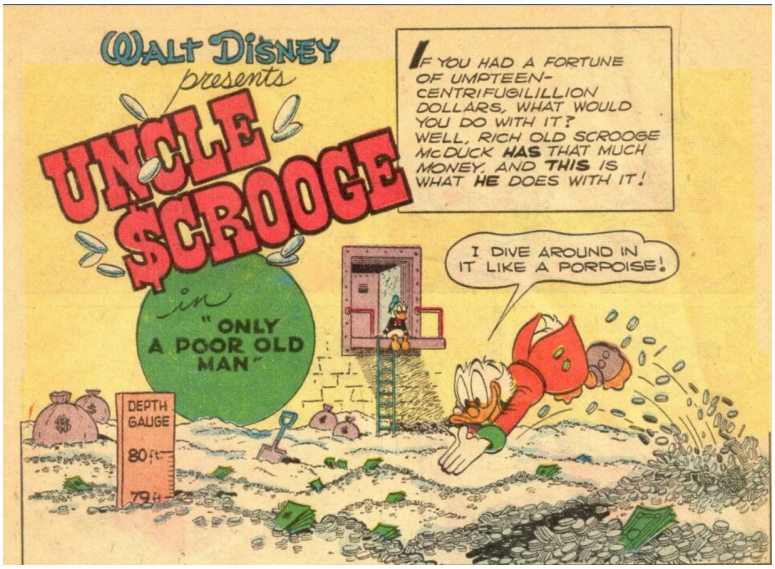

Figure 4: First panel of Carl Barks's first full Scrooge McDuck comic, "Only a Poor Old Man"

Start with the encoding itself. Imagine an image-generation model trained on a large collection of Disney comic books, including "Only a Poor Old Man," Carl Barks's first story with Scrooge McDuck as the protagonist.[100] When prompted with Scrooge's first line of dialogue—"I dive around in it like a porpoise."—the model generates a passable image of the story's first panel. (See Figure 4.)

The strongest version of the claim that generative-AI models are uninterpretable would be that Barks's artwork is not encoded in the model at all, because the model is unintelligible. Models are parameters—large collections of numbers.[101] These numbers bear no resemblance to "Only a Poor Old Man." If you printed out the parameters making up the model onto paper—enough pages to fill a decent-sized research library—no amount of squinting at them would make a visually recognizable Scrooge McDuck appear, like a Magic Eye diagram floating in space. Model parameters are not directly, literally intelligible to the human senses.

But that is the wrong test, because the mere fact that a model is encoded in a way that is not *directly* intelligible to the human senses is irrelevant. *All* digital media are encoded in ways that are not directly intelligible, twice over. Consider the PNG image file of the McDuck panel that we include in this essay, or the PDF version of our essay that you are currently reading. These, too, are large collections of numbers. File formats like PNG

100. *Four Color* #386 (March 1952).
101. *See supra* Part II; *See* Talkin', *supra* note 11, at 10–15 (discussing models).

and PDF—and others like JPEG, DOCX, and MP3—are not directly "recognizable" to a human, even if the bytes in them are literally written out on paper. This is an unremarkable observation in today's technology landscape. But we still speak, perfectly sensibly, about "viewing" a JPEG or "listening" to an MP3, because we can *make* them intelligible by using a computer to display or perform them. Copyright law recognizes that this decoding process can take place with "the aid of a machine or device."[102]

The same goes for physical devices. You cannot squint at the computer storage device on which a PDF is stored and read the document that way; if you use a scanning probe microscope to examine the patterns of electromagnetic charge in the device's semiconductors, it still will not look like anything familiar. But copyright law treats this device as a "copy" of the PDF, because it is a "tangible object" from which the work in the PDF can be made perceptible. The same is true of records (microscopic patterns of indentations on a vinyl disc), CDs (patterns of indentations on a reflective plastic disc), SSD drives (nano-scale patterns of electric charge stored in semiconductors), and much else. There is no question that these different physical formats can all constitute "copies" of a work, even though none of them is "recognizable" to a human without "the aid of a machine or device."[103]

This is a settled principle in copyright law.[104] In 1908's *White-Smith Music Publishing Co. v. Apollo Co.*, the Supreme Court held that player-piano rolls could not count as infringing copies, writing, "They are not made to be addressed to the eye as sheet music, but they form a part of a machine."[105] The very next year, Congress brought player-piano rolls inside the system of copyright law, giving copyright owners of music the exclusive right "to make any arrangement or setting of it or the melody of it in any system of notation or *any form of record* in which the thought of an author may be recorded and from which it may be read or reproduced" and imposing a royalty system for "the parts of instruments serving *to reproduce mechanically* the musical work."[106] This decision was carried forward into the current Copyright Act, which defines copies as "material objects . . . in which a work is fixed by any method now known or later developed."[107] This definition is used both in defining which works are "fixed" and thus copyrightable,[108] and also in

102. *See* 17 U.S.C. § 101.

103. *See id.*

104. Our thanks to Matthew Sag for helpful suggestions on this point.

105. White-Smith Music Publ'g Co. v. Apollo Co., 209 U.S. 1, 12 (1908).

106. An Act to Amend and Consolidate the Acts Respecting Copyright, Pub. L. No. 60–349, § 1(e), 35 STAT. 1075, 1075–76 (1909) (emphasis added).

107. 17 U.S.C. § 101.

108. 17 U.S.C. § 102(a) ("Copyright protection subsists . . . in original works of authorship fixed in any tangible medium of expression, now known or later developed, from which

defining when the copyright owner's exclusive rights to "reproduce the copyrighted work in copies"[109] and to "distribute copies . . . of the copyrighted work to the public"[110] have been infringed.[111] Copyright is technologically neutral; what matters is what can be done with a copy, not the details of how it is stored and encoded.[112]

Given this, there is no principled reason to say that, if memorized, encoding "Only a Poor Old Man" in the parameters of a generative model should not count as encoding it in the sense that is relevant for copyright. There is no difference in kind between the bytes that store a model file and the bytes that store a PDF file (except, perhaps, that a PDF happens to store one specific file, and a model stores transformations and copies of parts of potentially billions of files). There is no difference in kind between a USB drive storing a model and a USB drive storing a JPEG. It is only the relative novelty of generative-AI models (which are stored in file formats with names like safetensors and GGUF[113]) or perhaps the immense scale of models (which can run to trillions of parameters and require terabytes of storage), that makes them seem very different. The copyright system overcame its qualms about treating computer chips and player-piano rolls as tangible copies that can contain expressive works. It could overcome any similar qualms about generative-AI models if it wanted to do so.

Another version of the point has more force, and distinguishes models from JPEGs—to a degree. There is a standardized and widely implemented process to transform a JPEG-encoded file into a perceptible image on a computer screen. The process is nowhere near as simple as mapping each byte in the file to the color of a pixel on screen,[114] but it is unambiguous, efficient,

---

they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device"); 17 U.S.C. § 101 (definition of "fixed").

109. 17 U.S.C. § 106(1).

110. 17 U.S.C. § 106(3).

111. *Id.* § 101 (definition of "copies").

112. *See generally* Brad A. Greenberg, *Rethinking Technology Neutrality*, 100 MINN. L. REV. 1495 (2016) (discussing technological neutrality).

113. Vicki Boykis, *GGUF, the long way around* (2024), https://vickiboykis.com/2024/02/28/gguf-the-long-way-around/.

114. The JPEG standard (ISO/IEC JTC 1/SC 29/WG 10), designed by the Joint Photographic Experts Group (JPEG), is a method for compressing image data. The overarching goal is to reduce that amount of data (and thus storage space) that is needed to represent a particular image, without compromising (too much) the image's quality. There are three main steps to the JPEG algorithm: the Discrete Cosine Transform (DCT), (lossy) quantization of DCT outputs to lower-bit precision, and lossless encoding of the quantized outputs. This encoding (and the information needed to decode it) are formatted together into the bitsteam of the final JPEG file. Eric Roberts, *JPEG* (2024), https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/jpeg/

deterministic,[115] and requires no additional information from the user. If one has a large collection of JPEGs, they may be stored as files on a computer, or as individual entries in a database. In each case, it is straightforward to pick any individual JPEG out of the collection and make it visible. It is also possible to index a collection of files on a computer or database efficiently: start with the list of files, examine each one to see what it contains, and then store a short searchable abstract of those contents. In short, collections of JPEGs (and other familiar files) are *transparent* and *searchable*.

Architecturally, these facts derive from the way in which filesystems store items. In a typical filesystem, each file is stored in its own specific physical portion of the associated storage device. The bits that encode one JPEG are distinct from the bits that encode another. There is a data structure that describes how the files are stored; it is essentially an index that maps individual files to specific portions of physical storage. This means that individual files are physically and logically independent of each other.

A generative-AI model, on the other hand, can store the information it has learned from its training data in *partial* and *overlapping* ways. Any given parameter may contribute to the model's representation of numerous distinct concepts or correlations. Indeed, both the learning and generation processes propagate through the parameters in the model. In training, the model adjusts every parameter that contributed to an incorrect output. In generation, some parameter may contribute more in response to one input and less in response to another. But there is typically no master list of which parameters will contribute to which inputs, and no general way to restrict the processing only to those parameters that matter most.[116] There may be no "Scrooge McDuck" parameter in the comic-book model, no "Carl Barks"

index.htm (discussing the three main algorithmic steps of JPEG compression). Libr. of Cong., *JPEG Image Encoding Family*, SUSTAINABILITY DIGIT. FORMATS: PLAN. FOR LIBR. CONG. COLLECTIONS (May 8, 2024), https://www.loc.gov/preservation/digital/formats/fdd/fdd000017.shtml (defining the Library of Congress formal description of the JPEG digital format).

115. For most practical purposes, we can consider JPEG encoding and decoding to be deterministic. However, there can be slight differences between implementations of the algorithm, as well as small differences in rounding (due to lossy quantization to lower-bit precision) that can introduce small amounts of non-determinism.

116. Studying training-data influence and attribution remain active areas of research. They are often broadly grouped together with other techniques that study model interpretability. *See, e.g.,* Pang Wei Koh & Percy Liang, *Understanding Black-box Predictions via Influence Functions*, 70 PROC. MACH. LEARNING RSCH. 1885 (2017); Sung Min Park, Kristian Georgiev & Andrew Ilyas et al., *TRAK: Attributing Model Behavior at Scale*, *in* 202 PROC. 40TH INT'L CONF. ON MACH. LEARNING 27074—27113 (2023) (detailing influence and attribution estimation). *See infra* note 123 and accompanying text (discussing interpretability).

parameter, no "diving like a porpoise" parameter, and no "pixel # 3,881,308 from panel #3 on page #12" parameter.[117] Instead, the model's knowledge of *all* of these concepts—to the extent that it has any—is generally distributed across potentially a great many of its parameters. The content exists in the model's parameters, but this does not mean we have tools available that are guaranteed to tell us which specific model parameters encode it, or how.[118]

It is irrelevant that it is not always possible to describe an explicit one-to-one encoding or to pinpoint which bytes in a model file encode which particular works.[119] The Copyright Act's definitions of fixation and copies are functional, not formal. A work is fixed when its embodiment is sufficiently stable "*to permit it to be* perceived, reproduced, or otherwise communicated,"[120] and a copy is an object "from which the work *can be* perceived, reproduced, or otherwise communicated."[121] These capabilities do not depend on whether the work is encoded alongside or even overlapping with other works. "[N]o plagiarist can excuse the wrong by showing how much of his work he did not pirate."[122]

Nor does a generative-AI model build an index as it learns. The way in which each training example (potentially) modifies every parameter and the generation (potentially) depends on every parameter means that there is no simple concept of a "location" in a model to which an index entry could point. This trade-off is at the heart of Generative AI's power. By giving up on well-structured concepts and clearly definable relations between them, generative-AI models and algorithms are able to identify and imitate more subtle and complicated patterns in their training data. An image-generation model that generates an image of "coffee cat" is not simply adding together an image of "coffee" and an image of "cat"; it is drawing instead on a densely interconnected web of similarities and differences among numerous images of coffee and of images of cats, and among even more images of other things entirely.

---

117. There is a new area of active research that is trying to relate individual parameters (i.e., neurons in deep neural networks) to concepts like these. *See, e.g.,* Adly Templeton, Tom Conerly & Jonathan Marcus et al., Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet (2024) (unpublished manuscript), https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

118. *See supra* Part III.A.3 (distinguishing knowledge of memorization from the existence of memorization).

119. Our thanks to Eugene Volokh for helping sharpen this point.

120. 17 U.S.C. § 101 (emphasis added).

121. *Id.* (emphasis added).

122. Sheldon v. Metro-Goldwyn Pictures Corp., 81 F.2d 49, 56 (2d Cir. 1936).

Thus, generative-AI models are often neither transparent or searchable.[123] For the models of most interest today, there is no easy way to inspect their parameters and obtain a list of of all the information they have learned. Nor is it currently (or generally) possible to find "where" in a model a particular memorized example is encoded. If you do not already know that the first panel of "Only a Poor Old Man" is encoded in the comic-book model, there may be no straightforward way to find out whether it is. Even if you do know (or have strong reason to suspect) that the panel is encoded in the model, there may be no straightforward way to determine what prompts will cause the model to generate it. Nor is there a way to query the model for a list of all the panels it has learned, or the prompts that will generate them. In a sense, a large generative-AI model can be like Borges's Library of Babel: it contains literally incomprehensible immensities, to the point that it is extraordinarily difficult to index or navigate.[124]

C. Memorization and Compression

With this account of encoding in mind, consider again the claim that generative-AI models learn only "features," "patterns," or "statistical correlations." For example, Anthropic describes Claude's model(s) as follows:

> . . . Claude does not use its training texts as a database from which pre-existing outputs are selected in response to user prompts. Instead, it uses the *statistical correlations* gleaned from analyzing texts to construct a "model" of how language operates and what it means. The model represents those correlations in a series of numerical parameters (sometimes called "weights" and "biases") that enable software to generate sensible-seeming responses to requests from end users. Those parameters are what the model stores—not the texts of the training data.[125]

123. Memorization and *model interpretability* are two active fields of current research that study these questions. *See generally* Nasr, Carlini & Hayase et al., *supra* note 53 (for a recent large-scale measurement study on memorization in language models). *See generally* Templeton, Conerly & Marcus et al., *supra* note 117; Chris Olah, *Mechanistic Interpretability, Variables, and the Importance of Interpretable Bases* (2022), https://www.transformer-circuits.pub/2022/mech-interp-essay; Nelson Elhage, Neel Nanda & Catherine Olsson et al., A Mathematical Framework for Transformer Circuits (2021) (unpublished manuscript), https://transformer-circuits.pub/2021/framework/index.html (discussing interpretability).

124. *See generally* James Grimmelmann, *Information Policy for the Library of Babel*, 3 J. Bus. & Tech. L. 29 (2008).

125. *See* Response at 5–6, Concord Music Grp., Inc. v. Anthropic PBC, No. 3:23-cv-01092 (M.D. Tenn. Jan. 16, 2024) (internal citations omitted, emphasis added).

Similarly, Anthropic argues that Claude's model parameters contain "statistical correlations" that "... effectively yield 'insights about *patterns* of connections among concepts or how works of [a particular] kind are constructed.'"[126]

The idea here is that a "pattern" is an abstraction of some regularity in training data.[127] Instead of memorizing each individual training example, the model learns only the abstracted pattern—a more succinct description of some way in which multiple training examples resemble each other, or parts of an example are repeated. When invoked this way, the point of the argument is typically to contrast "learning" a "pattern" with "memorizing" "training data." It can be tempting to map this distinction onto other distinctions with legal significance, such as the line between uncopyrightable facts or ideas, and copyrightable expression.[128]

But this contrast can be misleading when used this way. The problem is that the "patterns" learned by a model can be highly abstract, highly specific, or anywhere in between. It is a "pattern" that every frame of 4K UHD video is 3840 pixels wide; a multi-modal model that learns to generate images of the correct resolution when the prompt contains "4k uhd" has not memorized any protectable expression. But it is also a "pattern" that in the first sentence of *The Restaurant at the End of the Universe* the word "widely" is followed by the word "regarded," and a model that learns enough such patterns can memorize all of Douglas Adams's oeuvre. So even to the extent that the results of model training are made up of "patterns," so is all memorization, because memorization is part of what happens during training.

A more accurate distinction would be that some learned patterns contain higher-order information that is not present in any single training example. Thus, for example, somewhere in a trained image-generation model's parameters, there may be information about the length of cats' whiskers gleaned from numerous pictures of cats. In contrast, for memorization, somewhere distributed among the parameters, there is a literal copy of a specific piece of training data—a specific picture of a cat. In this case, the pattern *is* the memorized training data. But to make this distinction is not to say anything new; it is simply to repeat the definition of memorization and to point out that not all learning is memorization.

Another popular comparison is to think of a generative-AI model as *compressing* its training data, discarding details while retaining more significant patterns. For example, one suit against Stability quotes statements made by then-CEO Emad Mostaque in 2022 to support the claim that "protected

126. *See id.* at 4–5 (emphasis added).

127. From here on, we will use "pattern" as shorthand for all of these phrases.

128. *See, e.g.,* Bracha, *supra* note 7 (describing patterns as uncopyrightable "meta-information").

expression from training images is copied, compressed, stored, and interpolated by diffusion models."[129]

The point of these metaphors is often the opposite of the previous one. Arguing that a model compresses its training data can be a way of arguing that it is an infringing copy of that training data. The data is still present in the model, just in a smaller version, like a thumbnail of a JPEG.[130]

But compression can be *lossy*: some of the information in the original data has been discarded and can no longer be reconstructed. But if compression is done well, the most important information will be retained and the least important discarded. The point of the JPEG standard, for example, is to retain the most visually significant information in the image, while discarding minor details that the human visual system is less likely to notice.[131] In the same way, even an immense generative-AI model may be much smaller than its training dataset. To the extent that the model can accurately summarize that dataset or successfully replicate the kinds of data in it, the training process has compressed the training data into a smaller but still useful version. The writer Ted Chiang has called ChatGPT a "blurry JPEG of all the text on the Web."[132] One point of the metaphor is that ChatGPT is trained on and compresses the text on the Web—like Soylent Green, ChatGPT is made out of people. Another point of his observation is that the compression is lossy: the JPEG is *blurry*. ChatGPT hallucinates, confabulates, and misquotes. In part because it compresses, the training process loses information, and fails to learn all that it hypothetically could.

We would refine Chiang's blurry JPEG compression analogy: not all parts of the JPEG are equally or completely blurry. Some training data are compressed more than others, and compression happens in different ways. Some of the compression is lossy: the information is discarded or trans-

129. *See* 1st Amended Complaint at 27, Andersen v. Stability AI, Ltd., No. 3:23-cv-00201 (N.D. Cal. Nov. 29, 2023) (Doc. No. 1) (capitalization removed); *id.* at 2 (quoting Mostaque as saying, "Stable Diffusion is the model itself. It's a collaboration that we did with a whole bunch of people . . . We took 100,000 gigabytes of images and compressed it to a two-gigabyte file that can recreate any of those [images] and iterations of those."); *id.* at 29 (quoting Mostaque as saying, "It's worth taking a step back and thinking about how crazy insane this is: we took a hundred terabytes of data—a hundred thousand thousand megabytes of images—2 billion of them—and we squished it down to a 2–4 gigabyte file. And that file can create everything that you've seen. That's insane, right? That's about as compressed as you can get.").

130. *Cf.* Perfect 10, Inc. v. Amazon. com, 508 F.3d 1146 (9th Cir. 2007) (holding that displaying a thumbnail of an image could infringe the display right in the image).

131. *See supra* notes 114–115 and accompanying text.

132. Ted Chiang, *ChatGPT is a Blurry JPEG of the Web*, New Yorker, Feb. 9, 2023, https://www.newyorker.com/tech/annals-of-technology/chatgpt-is-a-blurry-jpeg-of-the-web.

formed. But the portions that contain memorization are lossless: pieces of the training data are literally copied in the model, and, in the cases of regurgitation and extraction, can be retrieved in (near-)pristine condition at generation time.

These two sets of metaphors—about "patterns" and "compression"—are related. They are useful general ways to describe how generative-AI models work, but they do not tell us how a generative-AI model behaves in any specific instance, with respect to any specific training data. In the case of memorization, *the pattern is the details*. In other cases, higher-order patterns are leveraged to produce outputs that are not particularly similar to any specific training example.[133] And of course, there are gradations in between.

It is this complexity that is responsible for the generativity that is so compelling about generative-AI models, but this same complexity is difficult to grapple with directly. It is not just a matter of complex math that requires computer-science expertise to describe. Rather, even world-leading experts who are fluent in the math simply do not know all that much about the inner workings of models. Interpretability is a research challenge, not just a pedagogical one. As a result, it is often necessary to sidestep this complexity when talking about model behaviors.

D. Non-determinism and Generations

Another source of confusion about how models store information is that the generation process is typically *non-deterministic*—the same prompt can produce different outputs. For example, an LLM's parameters describe the strengths of connections between tokens in a sequence, not literally the sequence of tokens itself. Different generations, even starting from the same prompt, can activate these connections in different ways, leading the model to produce different outputs.[134] Thus, a generative-AI system might regurgitate its training data as output 10% of the time for a given prompt, and produce other outputs the other 90% of the time. For a different prompt, it might regurgitate training data 73% of the time. Even if the model produces a regurgitated output, it might not consistently regurgitate the same exact piece of training data.

---

133. This is an informal intuition for *generalization*, which we discuss below. *See infra* notes 161–162 and accompanying text. *See infra* Part III.F.

134. *See supra* notes 28–29 and accompanying text (discussing training, generation, parameters and tokens).

This nondeterminism might lead to skepticism that a model is indeed a copy[135] of training data: that even if the model sometimes regurgitates, it does not do so consistently, so there is no stable representation—no fixed copy—of the piece of training data inside the model. It also might lead to an intuition that regurgitation is itself a random process, so that similarity of an output to training data is a coincidence.

However, these intuitions rest on significant misconceptions about non-determinism in machine learning, specifically on what it means for parameters to model features of training data, and how these parameters are used to produce generations.[136]

Non-determinism is best understood as introducing subtleties that require courts and scholars to speak with care, and which may sometimes require them to draw legally significant lines along a continuum of model behavior. But the fact that generation is non-deterministic does not by itself have any necessary consequences for copyright law. Models can be copies of training data even if a prompt only sometimes extracts that data. At the end of the day, uncertainty about model behavior is an evidentiary issue, and mathematical tools can help in analyzing that evidence.

To clear these misunderstandings, in this section we discuss how non-determinism in generation does not contradict the fact that models memorize, and that memorized data is in models. (For simplicity, we primarily discuss LLMs, but similar points can be made about other types of models.) We make this argument in three parts. First, we briefly describe the role of non-determinism in computing, which we can more precisely discuss non-determinism during generation, and then explain its implications for memorization and copyright.

---

135. *See supra* note 63 and accompanying text (describing how, with respect to the definition of copies in copyright law, models that memorize training data are copies of that training data).

136. In this Essay, we focus on non-determinism in the generation process. Non-determnism also arises elsewhere in the generative-AI supply chain, especially in the training process. *See* A. Feder Cooper, Jonathan Frankle & Christopher De Sa, *Non-Determinism and the Lawlessness of Machine Learning Code*, *in* 2022 Proc. 2022 Symposium on Comput. Sci. & L. 1–8 (2022); A. Feder Cooper, Katherine Lee & Madiha Zahrah Choksi et al., *Arbitrariness and Social Prediction: The Confounding Role of Variance in Fair Classification*, *in* 38 Proc. AAAI Conf. on A.I. 22004–12 (2024); Jessica Zosa Forde, A. Feder Cooper & Kweku Kwegyir-Aggrey et al., Model Selection's Disparate Impact in Real-World Deep Learning Applications (2021) (unpublished manuscript). These other forms of non-determinism have significant social and legal implications, but they are beyond the scope of this Essay.

1. Non-determinism and Stochasticity in Computing

To start, let us be precise about why generation is non-deterministic. In computer science, an algorithm, piece of software, or system is *deterministic* when it always behaves in the same way when given the same input. A function that capitalizes a passage of text, for example, is deterministic: it always transforms the input `apple banana` into the output `APPLE BANANA`. In contrast, an algorithm, piece of software, or system is *non*-deterministic when the same input can cause different behavior, including different outputs. A non-deterministic function may change the capitalization of a passage of text differently each time it is run, for example, it might transform the input `apple banana` into `APpLE bANaNA` or into `appLe BanANa`.

Non-determinism may seem less intuitively useful, but it is an important part of computer science. Many real-world systems are so complex that they can only be modeled non-deterministically. A web server, for example, cannot predict exactly when user requests will arrive, or for which pages. The developers of the server must operate on the assumption that the rest of the Internet—all of the parts outside of their control—are essentially non-deterministic. Indeed, when developing one part of the server (e.g., the part that encodes pages for transmission to Internet users), they may need to model other parts of the server (e.g., the part that fetches data from storage) as non-deterministic—simply because it is too complicated to predict exactly which particular fetches will take more or less time, when they will complete, or if they will even complete at all, without errors. For another thing, nondeterminism can lead to more interesting—dare we say more creative—results. A text-to-image model that can produce four different variations from the same input prompt—and even more on demand—is more useful than one that can only use a given prompt to produce one specific image. For this reason, typical generation processes are specifically architected to be non-deterministic.

A non-deterministic system is also *stochastic* when its different possible behaviors for the same input can be described using the mathematics tools of probability.[137] In the example above, we said nothing about when or how often the function produces `APpLE bANaNA` or `appLe BanANa`. As we have described this function above, it is non-deterministic but not necessarily stochastic. But if it is the case that these two outputs—and any other possible capitalization—are, for example, equally likely, then the function is

137. More precisely, when they are described by a probability distribution over the outputs of a random variable. *See* Cooper, Frankle & De Sa, *supra* note 136 (detailing on how non-determinism and stochasticity are important concepts to understand when considering the legal implications of machine learning).

stochastic as well; we can model the behavior of the function's outputs using statistics.[138] In this case, we can use statistics to make useful, meaningful statements about the outputs for the input `apple banana`: how often `APpLE bANaNA` will appear, for example,[139] or whether an output with exactly three upper-case letters or one with exactly four is more likely.[140]

One might ask how we know that a non-deterministic system really is stochastic, as opposed to non-deterministic in a way that we cannot reliably model with tools from probability and statistics. The most satisfying answer is that it is stochastic because someone made it so. In this example, if the person who programmed the capitalization function made it explicitly random—if each letter is randomly and independently chosen to be lower-case or upper-case, with probability $\frac{1}{2}$ for each letter—then we can point to those random choices in explaining why the overall function behaves the way that it does. The probability that the output will be `APpLE bANaNA` follows from the individual probabilities that the first character will be `a` or `A`, that the second character will be `p` or `P`, and so on. Since each of these individual choices is stochastic, so is the overall behavior of the function. The power of statistics is that it lets us reason about the likely behavior of large and complex systems on the basis of the likelihood that their parts will behave in particular ways.

A reader who is accustomed to thinking of computers as reliable and consistent may at this point be wondering, *where does the randomness come from*? Don't computers always and only do what they are programmed to do, deterministically?[141] This is not the place to get into the philosophy of what randomness really is. Nor is it the place to explain the actual physical source

138. In this case, with equal probabilities for each output, we would say that the probability distribution over outputs is a "discrete uniform distribution." For another, simpler example, consider a fair, six-sided die: each outcome in $\{1, \ldots, 6\}$ is equally likely, with probability $\frac{1}{6}$ (leading to a total probability of 1), which we could also model with a discrete uniform distribution. However, both the function and the die could instead be implemented to make some outcomes more likely than others, in which case the outputs would would follow a "discrete categorical distribution." Continuing the die example, we could imagine that outcomes $\{1, 3\}$ each have probability $\frac{1}{4}$ and outcomes $\{2, 4, 5, 6\}$ each have probability $\frac{1}{8}$. (The total probability is still 1.) In both cases, we have probability distributions with statistical properties that we can leverage to reason about the behavior of outputs.

139. For the case in which each output is equally likely, this is once out of every 2^{11} times, on average.

140. Four. In this example, there are $2^{11} = 2048$ possible outputs; of these outputs, $\binom{11}{3} = 165$ have exactly three capitalized letters and $\binom{11}{4} = 330$ have exactly four capitalized letters.

141. And, indeed, what does this have to do with generative AI? We're getting there soon. *See infra* Part III.D.2.

of processes we typically think of as "random" (such as flipping a coin).[142] Instead, computers typically generate "random" choices (such as `a` or `A` in the example above) through a *pseudo-random* process: one that is in fact deterministic but whose behavior has as few predictable regularities as possible. They start with a *random seed*: an arbitrary number supplied from some external source, such as a user-supplied input or the exact time in nanoseconds between the arrival of network packets. They then use complicated (but deterministic) mathematical functions to turn the random seed into a sequence of other numbers, which appear random for all practical purposes.

The exact details are interesting, but not important for this Essay. What is important is that a computer's "random" values derive from a deterministic process that starts with an arbitrary random seed. The underlying process is actually deterministic, but it is stochastic for all practical purposes.[143] This is what allows computer scientists to implement statistical concepts in the code they write, and to use statistical tools to reason about the behavior of that code.

2. Stochasticity during Generation

Generative AI involves elements of both stochasticity and non-determinism—in model training, in deployed systems, and more.[144] The same is true of generation. It is a stochastic process:[145] the input prompt is transformed into an output generation in a way that depends on a large number of (pseudo-

142. Spoiler alert: it's quantum mechanics.

143. It is only deterministic if you know the random seed and are tracing through all of the computations based on it.

144. For example, training typically starts by initializing a model with random parameters, and continues by training the model on a randomly-chosen sequence of training examples—a fundamentally stochastic process. Repeating the training process with the exact same training data and exact same algorithm but a different random seed will produce a different model, one that may have quite different properties. *See* Cooper, Lee & Choksi et al., *supra* note 136 (for meaningful examples of these differences in a classification, as opposed to a generative, setting).

145. From a systems perspective, it can also be non-deterministic, if the generative-AI system contains multiple models (e.g., a Mixture of Experts); a given user request could get routed to a different model within the system, associated with no discernible statistical pattern, to produce the generation. We will stick to just considering stochasticity in the model in this discussion. *See* Maximilian Schreiner, *GPT-4 architecture, datasets, costs and more leaked*, THE DECODER (July 11, 2023), https://the-decoder.com/gpt-4-architecture-datasets-costs-and-more-leaked/ (discussing a rumour that OpenAI's ChatGPT-4 system contains 16 models to which user requests can get routed).

)random choices based on a random seed.[146] In an image diffusion model, the generation process starts by creating an image full of random noise, and then uses the prompt to guide the transformation of that random noise into a coherent image matching the prompt.[147] Repeating the generation process with the same prompt and a different seed means this de-noising process will start in a very different place and can end in one too—a completely different image that still reflects the prompt.

During generation for an LLM, random choices are involved when selecting which token to produce next in the output. As discussed in Part II, during generation the LLM takes the prompt, predicts the most likely next token in the sequence (based on the context of the prompt), generates that token as the next token in the sequence.[148] At each step, the model generates not a single next token, but instead its predictions of how likely *every* possible token (in the entire token *vocabulary*) is to follow the output so far. The overall system that is running the generation process through the model randomly selects one token from that distribution, favoring the possible tokens that the model predicts as more likely and disfavoring the ones that the model predicts are less likely, with what is more or less likely depending on the strengths of relationships between tokens that are encoded in the model's parameters.[149] The exact degree of favoring and disfavoring can also be adjusted at generation time by adjusting a setting called the *temperature*: loosely speaking, lower temperatures magnify the probabilities associated with high-strength relationships between tokens, while higher temperature discount those relationships and make next-token probabilities more random.[150] The process then repeats for the next token (with the previous

146. *See supra* notes 141–143 and accompanying text (describing random seeds and pseudo-randomness).

147. *See* Talkin', *supra* note 11, at 27–29 (discussing this process at a high level).

148. *See supra* notes 30–32 and accompanying text (discussing greedy decoding during generation).

149. *See supra* Part II.A.

150. In Part II.A, we discuss an example in which the most likely next token for `"once upon a"` would be `"time"`, assuming that the model is trained on a dataset that contains many fairy tales that use this phrase. To understand temperature a bit more and its relationship to the stochasticity of outputs produced by an LLM, let us now consider that the phrase is so common in the training dataset that there is a 95% probability that a language model trained on this dataset would generate the token `"time"` to follow the prompt `"once upon a"`. The other 5% probability is divided up among the other tokens in the whole token vocabulary used in model training (so that all token probabilities sum to 100%). In other words, this text example is not at all like the text-to-image case of `"cat in a red and white striped hat"` we discuss above, which could result in many possible different image generations; the near-surety of generating `"time"` as the next token means that this example is effectively (close to) deterministic.

one now looping back to serve as part of the input), and the one after that, and so on, so again there are a large number of random choices based on the initial random seed.

This is a very specific type of stochasticity, because of its dependence on the random seed. If you use the same random seed with the same prompt, the process is deterministic: you will get the same output.[151] But if you allow the random seed to vary (as seems to be the default mode of most generative-AI services), the process is stochastic. Each time you input a prompt, the system will use a different random seed for the generation process, and a different output will result. As Heraclitus said, you cannot step twice into the same river.

From this discussion, it follows that any claims one might want to make about how a generative-AI model behaves will be probabilistic. The question is not, "If I input prompt the model with X, will its output have property P?" Instead, the question is, "If I input prompt X, *what fraction of outputs* will have property P?" One way to find out is to tinker around: input the prompt, examine the output, and repeat a large enough number of times to arrive at a statistically meaningful conclusion. In theory, it might be possible sometimes to reason from first principles about how a model will behave. But in practice, the experimental method is currently the state-of-the art in making these kinds of claims.

Adjusting the temperature can change this behavior. Setting the temperature high flattens out the probabilities for the different tokens in the vocabulary: high temperatures make the probabilities for different tokens more equal so that, in (greedy) decoding, there is more randomness in determining the next generated token. (It increases the probability associated with tokens that have low probability at smaller temperatures, and decreases the probability for tokens that have high probability at smaller temperatures. For a vocabulary that has n different tokens, high temperatures effectively force the sampling of the next token to be like rolling an n-sided die.) Lower temperatures have the opposite effect; they reduce randomness, making high-probability tokens even more likely, and low-probability ones even less likely, than they are at higher temperatures.

Returning to our example, `"time"` does not originally have 100% probability assigned to it. Perhaps this is because Edgar Allen Poe's "The Raven" was in the training dataset; this poem has the opening line "once upon a midnight dreary," so there is some probability assigned to `"midnight"` as the next token. High temperatures would increase the probability for `"midnight"`, while low ones would reduce it. *See* Geoffrey Hinton, Oriol Vinyals & Jeff Dean, Distilling the Knowledge in a Neural Network (2015) (unpublished manuscript), https://arxiv.org/abs/1503.02531 (for background on temperature).

151. Again, we note this is true only with respect to the model. Once a model is embedded in a system that contains other types of non-determinism, this reproducibility of the same output for the same prompt may not be guaranteed. *See supra* note 145 and accompanying text.

We now bring things back to memorization. A claim about extraction or regurgitation will typically take the following form: a model, when (a) given a particular type of input, will (b) produce a particular type of memorized output, (c) *with a particular probability*. That probability could be .01 (i.e,. a 1% chance), it could be .99 (i.e. a 99% chance), it could be some value in between, or it could be even more extreme. The issue for copyright law, then, is what to do with this knowledge, and in particular, what to do with the fact that element (c)—the probabilistic element—is inescapable.

3. Consequences for Copyright

Consider first the question of whether a model is a copy of a training work. One possible conclusion is that that the specific probability of regurgitation is mostly irrelevant, and that what matters for copyright purposes is that there is or can be any meaningful probability of regurgitation at all.[152] On this view, a model is a copy of a work as long as there is any realistic possibility that the work "*can* be perceived, reproduced, or otherwise communicated" from it.[153] The Copyright Act uses the word "can," and even a small probability is enough to establish the possibility. If nothing else, one could repeat the generation process (with a fresh, arbitrarily chosen random seed each time) until a near-exact copy emerges.[154]

There is something to this view. In particular, the fact that there is some probability that the generation process could produce a different output should not by itself mean that a model has not memorized training data. Imagine a jukebox. When the user inserts a quarter, half the time the juke-

152. Our asides on temperature should help make this point clear. Consider using a prompt with low temperature, which adheres to the strengths of the relationships between tokens that are encoded in the model. Regurgitation in this setting makes clear that memorization is in the model—it is a direct product of the learned relationships. But using the same model and same A user could set the temperature so high that the response to the same prompt is effectively completely random, and contains not regurgitated data at all. In both cases, the model is the same; how we ran it is different. This has a clear impact on how much memorization is surfaced at generation time, but it would make no sense to say that this changes how much memorization is in the model. How one uses the model—with different temperature settings, prompts of varying lengths, etc.—plays a role in specific probabilities for regurgitation, but should not be confused more generally with there being a meaningful probability to regurgitate at all. *See supra* note 150 and accompanying text (discussing temperature). *See* Carlini, Tramèr & Wallace et al., *supra* note 54, at 2368–40 (describing experiments that measure memorization in LLMs that examine the role of temperature).

153. 17 U.S.C. § 101 (emphasis added).

154. One might call this approach "adversarial," but we think that overstates the case. It is akin to rolling a die until it lands on 6. *See infra* Part III.H.

box plays the selected record. The other half the time, it eats the quarter and does nothing. It would be absurd to say that the jukebox does not contain a copy of the sound recording on the record simply because there is only a 50% chance of playing the record each time the jukebox is used.

Indeed, even "deterministic" processes have a little non-determinism baked in to them. A jukebox in factory condition has a small but non-zero probability of malfunctioning: maybe there will be a power surge at exactly the wrong moment and the tonearm will never make contact with the record. The algorithm for converting an MP3 file to an audio signal is deterministic—but there is always a non-zero (if tiny) probability that cosmic rays will corrupt the computer's memory in a way that overwrites the audio data with noise. What matters is only that the probability is high enough that it reliably works enough of the time—what matters is a question of degree.

Still, we have used phrases like "high enough," "realistic," and "meaningful" because a threshold of any non-zero probability is too low. Consider an image generation model that outputs a completely random image, or an LLM that outputs a completely random string of tokens. Each of these models has a non-zero probability of outputting *any* output of the specified size, including any arbitrarily-chosen copyrighted work. A monkey at a typewriter hitting keys at random has an almost unimaginably small probability of generating the text of *A Tale of Two Cities*—but the probability is still non-zero.[155] Given enough monkeys, enough typewriters, and enough time, they will generate *A Tale of Two Cities* even though neither the monkeys nor the typewriters have memorized it. So too with a generative model and memorization: the probability of generating a near-exact copy of a piece of training data has to be high enough to rule out coincidence. And indeed, it is essentially impossible that it is a coincidence that machine-learning researchers are able to reliably extract such copies at high rates.[156]

As for infringing outputs, the probability of particular kinds of outputs bears on how common and uncommon particular uses of a system are. In the same way that it is an empirical question what the fraction of infringing works on a file-sharing service is, or what the fraction of infringing links for a particular search query is, it is an empirical question what the fraction of infringing outputs for a particular prompt is. It is a question of copyright

155. "*Mr. Burns*: This is a thousand monkeys working at a thousand typewriters. Soon they'll have written the greatest novel known to man. Let's see. It was the best of times, it was the blurst of times?! You stupid monkey!"

156. *See* Carlini, Tramèr & Wallace et al., *supra* note 54; Carlini, Hayes & Nasr et al., *supra* note 54; Carlini, Ippolito & Jagielski et al., *supra* note 52; Nasr, Carlini & Hayase et al., *supra* note 53.

law and policy what the consequences of this empirical finding are.[157] It almost certainly matters in the fair-use analysis whether common prompts yield memorized outputs 5% or 50% of the time, and it likely also matters for remedies, such as the size of damage awards and whether to issue an injunction. But these are complex balancing tests, and it is not at all obvious how any particular probability for any particular prompt should matter. These probabilities are important, relevant evidence, but they do not by themselves determine any legally relevant lines. Courts will need to do that, and they may well draw different lines in different contexts. Nevertheless, *our* bottom line is that non-determinism is an important phenomenon that courts should take into account, but does not dictate any particular view of the copyright consequences of regurgitated generations.

E. How Much Memorization?

Having discussed how memorization is encoded as copies inside of generative-AI models, let us now consider the question of *how much* these models memorize. Some plaintiffs and scholars argue that generative-AI models *only* memorize their training data; some defendants and scholars argue that generative-AI models *never* memorize. The truth lies somewhere in between. Some (but not all) of the learning that generative-AI models do qualifies as memorization. The question of how much a model memorizes is an empirical one, which cannot be answered except with reference to a specific model and specific ways of identifying what it has memorized. That said, there is suggestive evidence that at least some memorization is normal behavior for a generative-AI model that is powerful enough to be useful.

First, note that there are generative-AI models that memorize *nothing* in their training data. Consider a model that is trained on an empty dataset, where its parameters are initialized to random numbers. Its parameters will have the same values they have at the start of the training process: random numbers. The model has memorized absolutely nothing, and there is no way to extract training examples from it. Similarly, note that there are generative-AI models that memorize *everything* in their training data. Consider an image-generation model that is trained exclusively on millions of copies of the first panel of "Only a Poor Old Man" (Figure 4). Assuming the model is large enough, its parameters will be exquisitely tuned to generate the panel. The model will be able to reconstruct the panel perfectly.

Of course, both of these models are almost completely useless. The model with random parameters is capable of generating nothing coherent; the specialized model is capable of generating one coherent output. If you

157. *See infra* Part II.E; II.H (discussing such empirical questions).

want random outputs, or you want the first panel, these models will do, but if that was what you wanted, there were easier ways to achieve these outcomes. To be fair, we did not say these were *good* models—but they are generative-AI models all the same. Nothing in the nature of a generative-AI model inherently requires or prohibits memorization. Everything depends on how it is configured and trained.

Machine-learning researchers have developed several different ways of attempting to quantify the amount of memorization in a model. For text generation, one approach is to prompt an LLM with a random, contiguous portion of a randomly selected training example, to see whether the model responds with an output that completes the rest of the example from which the prompt was taken.[158] By this method, for example, the 6-billion-parameter GPT-J model memorizes at least 1% of its training dataset.[159] This common measurement procedure, which relies on a relatively simple prompting strategy, likely significantly underestimates the total amount of memorization in a model.[160]

The key capability that makes a model useful is *generalization*: its ability to perform well on unseen data.[161] A generative-AI model generalizes well when it produces sensible generations in response to previously unseen prompts—i.e., outputs that are not *just* copies of their training data inputs. Researchers have also developed a circumstantial but suggestive case that the quality of a model is partly dependent on memorization—that some amount of memorization might even be *required* for effective generalization.[162] By

158. Carlini, Ippolito & Jagielski et al., *supra* note 52.

159. *Id.*

160. Carlini, Tramèr & Wallace et al., *supra* note 54, at 11 (discussing similar methodology with respect to measuring memorization in GPT-2) ("The important lesson here is that our work vastly *under-estimates* the true amount of content that GPT-2 memorized There are likely prompts that would identify much more memorized content, but because we stick to simple prompts we do not find this memorized content.").

161. GenLaw Glossary, *supra* note 51 ("Generalization in machine learning refers to a model's ability to perform well on unseen data, i.e., data it was not exposed to during training. Generalization error is usually measured evaluating the model on training data and comparing it with the evaluation of the model on test data.). Devising useful metrics for generalization is also an active area of ML research. Chiyuan Zhang, Samy Bengio & Moritz Hardt et al., *Understanding deep learning (still) requires rethinking generalization*, 64 Commc'ns ACM 107–115 (2021).

162. Congzheng Song, Thomas Ristenpart & Vitaly Shmatikov, *Machine Learning Models that Remember Too Much*, *in* 2017 Proc. 2017 ACM SIGSAC Conf. on Comput. & Commc'ns Sec. 587–601 (2017); Satrajit Chatterjee, *Learning and Memorization*, *in* 80 Proc. 35th Int'l Conf. on Mach. Learning 755—763 (2018); Vitaly Feldman, *Does learning require memorization? a short tale about a long tail*, *in* 2020 Proc. 52nd Ann. ACM SIGACT Symposium on Theory Comput. 954–959 (2020); Vitaly Feldman &

one estimate, only 0.1% of some large language models' overall parameters contain verbatim memorization; for other models, this number is 10%.[163] The details depend heavily on implementation decisions, but within a given model family, larger models tend to memorize more than smaller models.[164] Examples that are duplicated in the training data—and hence trained on more often—are more likely to be memorized.[165]

It makes intuitive sense that memorization is a Goldilocks phenomenon; models are most useful when they memorize just the right amount, neither too little nor too much. On the one hand, memorizing the alphabetical list of the fifty U.S. states is a feature, not a bug; a model that confidently inserts Cahokia and West Dakota into the list of states might charitably be described as "hallucinating" or "garbage." On the other hand, a model that *only* memorizes is just a glorified (or perhaps subpar) search engine.

Chiyuan Zhang, What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation (2020) (unpublished manuscript), https://arxiv.org/abs/2008.03703; Chiyuan Zhang, Samy Bengio & Moritz Hardt et al., *Identity Crisis: Memorization and Generalization Under Extreme Overparameterization*, *in* 2020 Int'l Conf. on Learning Representations (2020); Gerrit van den Burg & Chris Williams, *On Memorization in Probabilistic Deep Generative Models*, *in* 34 Advances Neural Info. Processing Sys. 27916—27928 (2021) (studying memorization and generalization in deep learning). *See infra* Part III.F (discussing generalization and learning beyond memorization).

163. Lee, Ippolito & Nystrom et al., *supra* note 72, at 7 (citing 1% memorization in a 1.5B parameter model similar to GPT-2, and 0.1% memorization of the same architecture trained on a deduplicated version of the dataset). Carlini, Ippolito & Jagielski et al., *supra* note 52 (for similar results finding 1% memorization). Nasr, Carlini & Hayase et al., *supra* note 53, at 15 (discussing extent of memorization in the GPT-Neo 6B model). These numbers serve as examples of measuring particular types of memorization under certain conditions and for specific models. They should not alone be taken as a general claims about all models. The nuanced relationship between model capacity and memorization is not entirely understood.

164. For example, in Meta's Llama family of models, Llama-65B (which has 65 billion parameters) memorizes more than Llama-7B (which has 7 billion parameters). Nasr, Carlini & Hayase et al., *supra* note 53; Carlini, Ippolito & Jagielski et al., *supra* note 52; Carlini, Tramèr & Wallace et al., *supra* note 54. Another observation is that a model that is much smaller than the dataset it is trained on cannot effectively memorize everything in that dataset (though it could memorize some of the dataset). The compression analogy helps us (roughly) understand this point: larger models have more storage capacity than smaller ones; a smaller model has less space to contain high-fidelity compressions of its training data. *See supra* Part III.B.2 (discussing the blurry jpeg analogy).

165. *See supra* note 72 and accompanying text; Lee, Ippolito & Nystrom et al., *supra* note 72 (discussing deduplication).

F. Learning Beyond Memorization

As should hopefully be clear, memorization is not interchangeable with learning. The definition of "memorization" we are using refers to near-exact reproduction of a substantial piece of training data.[166] This is a much narrower concept than the kinds of learning and generalization that a model may be capable of. For some modalities (e.g. images), the definition excludes exact reproduction of small sub-portions of training examples.[167] It also excludes generalization from patterns present in many training examples. Both of these are learning but not memorization.

Some critics of generative-AI have tried to deny that there is a meaningful difference. They argue that all of Generative AI is a mosaic or collage; it consists of rearranged pieces drawn from training data. This is a misleading picture, because it ignores the possibility of generalizing from statistical information[168] in the model that has been synthesized from training on large amounts of diverse data.[169] An AI-generated image from Midjourney is not a Frankenpicture of sewn-together exact copies of fragments of existing images; the learned concepts stored in Midjourney's model are at much higher levels of abstraction than individual pixels. Nor is this image simply borrowing these concepts—symmetrical composition, the iridescence of a mollusk's shell—from individual images; many or most of them will be generalizations from numerous training examples. There is a sense in which one could describe *Infinite Jest* as a collage of words drawn from other books: a "the" from *Moby Dick*, a "woman" from *The Feminine Mystique*, a "who" from *Horton Hears a Who*, and so on. But in another, more accurate sense, this is not what is going on at all, and "collage of individual words" completely fails to describe any book's relationship to the rest of literature. And so on. It is precisely because *not* all learning is memorization that memorized training data meaningfully stick out.

On the other hand, Oren Bracha gives an argument from copyright theory that any learning performed by a generative-AI model consists of a "process of extraction of metainformation from expressive works that then

166. *See supra* Part III.A.

167. Depending on measurement choices, it also could exclude exact reproduction of an entire training-example image within a generation—e.g., a generation of a living room scene that contains a painting on the wall that replicates one of Kerry James Marshall's works.

168. Again, it is the case that some of this information is memorization—is literal copying—but not all of it. *See supra* Part III.B.2.

169. Talkin', *supra* note 11, at 63; Cooper, Lee, Grimmelmann & Ippolito et al., *supra* note 26, at 38.

enables the production of new and different expression(s)."[170] In his view, this "[m]ere physical reproduction, delinked from enjoyment of the expressive value of a work and completely incidental to accessing unprotected meta-information, is categorically beyond copyright's domain."[171] In other words, Bracha identifies learning in a model with uncopyrightable ideas, and locates expression only in the model's outputs.

In our view, memorization refutes this interpretation of how generative-AI models work. When a model regurgitates an expressive work, the model's parameters are not "delinked from enjoyment of the expressive value of a work" and certainly do not contain only "meta-information."[172] There is a straightforward causal connection from the (expressive) training data through the model to the (expressive) output, even if we do not have the tools to directly pinpoint the links along the path.[173] Either the model contains the work's expression, in which case the legal argument fails, or it does not, in which case the reappearance of the exact same expression in the output is a (fantastically improbable) mystery.

Bracha's stronger argument is that learning should be regarded as a case of merger: the memorization of (some) expression is noninfringing "to the extent necessary for accessing the unprotectable material" that consists of the larger patterns across many works.[174] This claim, of course, depends on the degree to which memorization really is "necessary" to extract these larger patterns, which, as we have noted, is a difficult and contested research question.[175]

---

170. Bracha, *supra* note 7, at 8; *see also* Christopher J. Sprigman, *Upsetting Conventional Wisdom of Copyright Scholarship in the Age of AI*, Jotwell (Mar. 28, 2024), https://ip.jotwell.com/upsetting-conventional-wisdom-of-copyright-scholarship-in-the-age-of-ai/ (reviewing Bracha's draft).

171. Bracha, *supra* note 7, at 24.

172. In our view, the term "meta-information" here can be misleading in the same way that "feature," "pattern," and "statistical correlation" can have multiple meanings at different levels of abstraction. It is possible that much of the information in a trained model reflects higher-level information that cannot be easily pinned down as expressive; however, as we discuss above, memorization means that some of this information (these features, patterns, statistical correlations) are literal copies. *See supra* Part III.C.

173. *See supra* Part III.A.3.

174. Bracha, *supra* note 7, at 25.

175. There are also some doctrinal challenges with this approach, most notably the degree to which merger can be asserted as a defense to the *defendant*'s otherwise infringing behavior, rather than being a limitation on copyrightability or an argument that the *plaintiff* has too thoroughly interwoven idea and expression to separate them. *See generally* Pamela Samuelson, *Reconceptualizing Copyright's Merger Doctrine*, 63 J. Copyright Soc'y U.S. 417 (2016) (discussing merger doctrine).

Finally, the boundary of what counts as memorization is necessarily vague. We have been using terms like "near-verbatim," "small," and "many" without trying to make them precise. Different machine-learning researchers could (and do) quite reasonably use different metrics for these ideas. Indeed, one of the crucial theoretical underpinnings of ML research is that any such measurable quantity—similarity, frequency, size, etc.—can be reasoned about abstractly. For example, one can describe an algorithm that depends on a measure of similarity (or "distance") between two examples, without specifying which measure one is using. To implement the algorithm, one must first pick a metric to use (e.g., to measure the similarity of two passages of text by counting their differences letter by letter), and then typically also pick thresholds (e.g., one passage is a "near-verbatim" copy of another when their differences are less than 5% of their total length).

Drawing a line between learning and memorization requires making technical choices of this sort, and any such line is inherently arbitrary. It may be necessary to draw a line, and some choices may be more useful than others, but at the end of the day, memorization is one extreme on a continuum of ways to learn, not a discrete category. For these reasons, it is also hard to draw a firm line like Bracha does between "meta-information" and expression for Generative AI. Expression and information can be transformed during learning, but they can also be copied directly into model parameters—and the amount that one deems "copied" depends on one's chosen metric for memorization.[176]

G. Models are not VCRs

In the preceding sections, we have made clear that all trained generative-AI models memorize, that useful generative-AI models also generalize, and that distinguishing between memorization and generalization is neither simple nor straightforward. Nevertheless, some defendants in generative-AI copyright-infringement lawsuits have claimed that the memorization demonstrated with their systems is not "typical,"[177] but is instead an "unintended occurrence."[178] The implication is that, even if memorization is found to be infringing, generalization is the intended, main, non-infringing use.[179]

176. *See supra* notes 51–55 and accompanying text (discussing different memorization definitions and metrics in the technical literature).

177. *See infra* note 200 and accompanying text.

178. *See infra* note 214 and accompanying text.

179. From this basis, many responses from defendants then lay responsibility for extracting memorized training with "adversarial" end users, which we discuss in the following section.

In a similar vein, some AI companies and observers have argued that generative-AI models are general-purpose copying technologies, like VCRs. As such, they argue that the substantial-noninfringing-use doctrine from *Sony Corp. of America v. Universal City Studios, Inc.* is a good fit for Generative AI.[180] In *Sony*, a group of entertainment companies sued Sony for copyright infringement, arguing that consumers used Sony VCRs to infringe by recording programs broadcast on television. The Supreme Court held that Sony could not be held contributorily liable for infringements committed by VCR owners. "[T]he sale of copying equipment . . . does not constitute contributory infringement if the product is . . . capable of substantial noninfringing uses."[181]

The *Sony* doctrine is appealing because generative-AI models are dual-use technologies. They can be put to infringing uses: a user could coax a model to produce verbatim copy of a particular Scrooge McDuck cartoon from that model's training data, or use an LLM to grammar-check an infringing sequel to a popular novel. But they can also be put to non-infringing uses: many outputs from generative-AI models are generalization: expressive works that are not substantially similar to any already-existing expressive works. The *Sony* doctrine provides a bright-line rule that allows dual-use technologies to continue to be available for these beneficial non-infringing uses.[182] As Microsoft put it in moving to dismiss a copyright lawsuit from *The New York Times*, "copyright law is no more an obstacle to the LLM than it was to the VCR (or the player piano, copy machine, personal computer, internet, or search engine)"—all dual-use technologies.[183]

The fact that memorization is in the model, however, makes us skeptical about the VCR analogy, for two reasons. The first is formal: U.S. copyright law uses the physical fact of copying to distinguish between direct and secondary liability, so memorization in a model can affect whether *Sony* applies at all. The second is functional: a VCR is completely neutral among different expressive works, whereas a model is typically better at generating some specific works than others.

We start with the formal analysis of who makes the relevant copies.[184] In *Sony*, it was clear that the users were the direct infringers (if anyone was).

180. Sony Corp. of Am. v. Universal City Studios, Inc., 464 U.S. 417 (1984).

181. *Id.* at 442.

182. *See generally* David A. Widder, Helen Nissenbaum & James Grimmelmann, Moral and Legal Responsibility for General-Purpose Technologies (2024) (unpublished manuscript, on file with with authors).

183. Memorandum of Law in Support of Motion at 2, The N.Y. Times Co. v. Microsoft Corp., No. 1:23-cv-11195 (S.D.N.Y. Mar. 4, 2024) (Doc. No. 65).

184. *See* Am. Broad. Cos., Inc. v. Aereo, Inc., 573 U.S. 431, 453 (2014) (Scalia, J. dissenting) ("the question is *who* does the performing") (emphasis added).

Sony sold a device that could be used by others to make copies of the plaintiff's works; it made no copies itself. Sony could be liable, if at all, only secondarily, and the Supreme Court focused on contributory infringement. Contributory infringement requires at least knowledge of the infringement, and one reading of *Sony* is that this knowledge will not be attributed to the defendant on the basis of a generalized awareness that the device could be used to infringe, so long as the device can also be used in non-infringing ways.

But direct infringement liability for the person who actually makes the infringing copy is strict, regardless of whether they intended to infringe, or knew that they might be infringing. The *Sony* defense has never shielded direct infringers. To the extent that an AI company creates a model that is found to be an infringing copy of a work in the training dataset,[185] that is formally direct infringement, not contributory, and *Sony* does not apply. Further, to the extent that the model itself is an infringing copy of training data, anyone who copies the model is also a direct infringer unprotected by *Sony*. Indeed, to the extent that a generative-AI system produces an infringing output because a model embedded within it has memorized a work and is now regurgitating it, the provider of that system might also still be a direct infringer and outside of the *Sony* rule.

Our point here is not that this is a good or bad outcome; we take no position on whether *Sony* or something like it should apply as a policy matter. Instead, our point is that U.S. copyright law is currently deeply committed to a formal and technically searching analysis of which tangible objects are copies and who is responsible for making tangible objects into copies. A VCR is not a copy of a movie or TV program; it is a device that can be used to make copies of them. But a generative-AI model that has memorized training data is a copy of that training data.[186] It can also be used to make further copies of that training data (and of other works, depending on the prompt). But the fact that it can be used to make copies on the back-end (i.e., generations at generation time) does not avoid the fact that, due to memorization, it is itself a copy on the front-end (i.e., the model, as a result of training).[187] In other words, memorization plays a crucial role in determining whether *Sony* is the doctrinally appropriate category with which to analyze generative-AI models. Copyright law does not necessarily need to work this way, but if it does, memorization matters.

---

185. For readers with a computer science background, we emphasize again that a model *containing* a copy of *part* of a work due to memorization is still a "copy" of the work in the sense that copyright law uses the term "copy." *See supra* note 63 and accompanying text.

186. *See supra* note 185 and accompanying text.

187. *See supra* notes 58–59 and accompanying text (describing this front-end/back-end framing).

The second reason that memorization matters to the VCR analogy has to do with the effective capabilities of a model. A VCR is almost entirely content-neutral. It can be used to play or record any audiovisual work, limited only by the fidelity and length of the tape. A Sony Betamax functioned identically when recording *Bride of Frankenstein* (prohibited by Universal) and recording *Mr. Rogers' Neighborhood* (encouraged by Fred Rogers). It is a general-purpose tool. The same goes for other copying technologies that *Sony* has been applied to, in court or in policy arguments, including photocopiers, the personal computer, and Internet service. A photocopier does not distinguish between *War and Peace* (public-domain) and *Things Fall Apart* (under copyright).

But a generative-AI model that regurgitates training data is emphatically not neutral with respect to copyrighted works. It is capable of outputting some works but not others. As discussed above, what makes memorization distinctive is that the model stores near-exact copies of portions of works it was trained on. To the extent that these memorized potions of works can be regurgitated, extracted, or reconstructed, that is a real-world difference between works the model memorized from its training data, and works it was not trained on or did not memorize. The model behaves differently towards some works than others.

This is a genuine functional difference between VCRs and generative-AI models, and it goes to the heart of what makes Generative AI so powerful. Generative-AI models engage with the *content* of expressive material. That is why they are able to engage in so many tasks that were previously considered to be human-only: they are capable of imitating and modifying creative works in ways that seem to an observer to have content and meaning. It explains both the hype and the hatred that Generative AI inspires. To reduce a generative-AI model to merely a copying technology, like a photocopier or VCR, is to overlook its most distinctive feature.

H. "Adversarial" Users

Defendants tend to lay the responsibility for regurgitating copyrighted expression with "adversarial" users. They argue that plaintiffs' examples of regurgitation only arise because the plaintiffs used atypical or "adversarial"[188] prompting strategies that no typical or "normal"[189] user would reasonably use in practice. If one were to accept the (incorrect) analogy that models are

188. OpenAI, *OpenAI and journalism* (Jan. 8, 2024), https://openai.com/blog/openai-and-journalism.

189. Response at 4, Concord Music Grp., Inc. v. Anthropic PBC, No. 3:23-cv-01092 (M.D. Tenn. Jan. 16, 2024).

like VCRs,[190] then these "adversarial" users would be like bootleggers that use VCRs to produce unauthorized copies. In these lawsuits, these users are often the plaintiffs themselves, who have used the defendants' systems to extract their own copyrighted works. Thus, the argument goes, these examples of regurgitation should be disregarded.

We do not believe that adversarial usage can be so easily disregarded. First, "adversarial" users can only extract memorized content if the model has memorized this content in the first place. Second, the line between "adversarial" usage and "typical" usage is not fixed or stable—and even if a line can be drawn, the relative balance of the two can also vary. And third, AI-system creators have the ability to anticipate some "adversarial" usage and adopt safeguards against it. We take up the first two arguments in this section, and discuss system-level safeguards in the next.

To repeat, regardless of whether a user is "adversarially" trying to extract memorized training data or just happens to do so accidentally, it is only possible to generate memorized training data if that data is encoded in the trained model.[191] Indeed, as we have discussed, the fact that a user can use a detailed prompt to extract a specific memorized training example is an unsurprising consequence of how generative-AI training works.[192]

Consider an LLM. During training, a generative-AI model learns features—certain "statistical correlations"[193]— from its training data. In the case of an LLM, these correlations are patterns in the natural language in its training dataset. The trained LLM can then be used to generate natural-language text; it takes a text prompt as input and emits as output a continuation, or *completion*, of the prompt. Crucially, the model predicts which of many possible completions is "most likely" based on the statistical patterns it has learned about language from the data on which it was trained.[194] If the model regurgitates training data in response to a given prompt, it does so *because it has learned* that the example's text is the most likely completion for

190. *See supra* Part III.G.

191. *See supra* Part III.A.2.

192. *See supra* note 59 and accompanying text (quoting the technical literature on the relationship between prompt length and successful extraction).

193. Response at 4–6, *Concord Music Grp., Inc. v. Anthropic PBC*, No. 3:23-cv-01092.

194. This is one of the intuitions behind why duplicated training examples in the training dataset result in models exhibiting higher levels of memorization: an example that appears multiple times in the training dataset can seem like a "more likely" language pattern. Lee, Ippolito & Nystrom et al., *supra* note 72. The *Times* alleges that OpenAI's training process samples "higher-quality" sources, including *Times* articles, more frequently during training. Complaint at ¶ 90, N.Y. Times Co. v. Microsoft, No. 2:24-cv-00711 (C.D. Cal. Dec. 27, 2023). *See supra* note 72 and accompanying text (discussing deduplication).

the given prompt.[195] Of course, the prompt plays an important causal role in actually eliciting this behavior. But before the prompt is entered, the model has, latent within it, learned "statistical correlations" that happen to reflect memorization of some of the training data.

We can revisit the *New York Times*'s complaint against OpenAI in light of this discussion. Recall that the *New York Times* was able to prompt ChatGPT to produce lengthy near-verbatim excerpts from specific *Times* articles, which the *Times* then cited in its complaint as proof of infringement. The *Times* prompted ChatGPT with long-sequence text prefixes from its articles; in some cases, based on this context, ChatGPT would generate the corresponding suffix—text that completed the remainder of the article excerpt. (See Figure 1.)

OpenAI argued in its public response that "It seems they intentionally manipulated prompts, *often including lengthy excerpts of articles*, in order to get our model to regurgitate."[196] But the fact that the *Times* could *cause* ChatGPT to regurgitate articles does not answer the question of whether OpenAI should or should not have trained more or otherwise modified ChatGPT in a way that makes regurgitation *possible*. It is not a foregone technical conclusion that prompting with "lengthy excerpts of articles" should necessarily lead to the rest of the article being surfaced by either the model or system in which it is embedded.[197] By itself, regardless of user intent, regurgitation is a

195. As always, the technical details introduce further complications. First, the generation process typically involves an element of randomness, and so the same prompt can yield different generations. Second, software-engineering and systems-implementation decisions can affect how a model behaves. For example, it is unclear why prompting the ChatGPT system to repeat the same token forever (e.g., `"poem"`) causes the model to "diverge" from behaving like a conversational chatbot and to produce (sometimes very long) sequences of seemingly arbitrary training examples. Nevertheless, our simplification serves as a useful mental model for what happens when memorized training data is extracted. *See supra* notes 30–32 and accompanying text (discussing randomness in sampling the next token). *See* Nasr, Carlini & Hayase et al., *supra* note 53 (discussing divergence and extraction in ChatGPT).

196. OpenAI, *supra* note 188 (emphasis added). It should not be surprising that these long-context prompts could extract *Times* articles. OpenAI had trained this version of ChatGPT on *Times* articles, and so prompting with a long sequence of article text (in some sense) encouraged or guided the model's next-token generation process to complete the rest. Carlini, Ippolito & Jagielski et al., *supra* note 52, at 5 (" . . . conditioning a model on 100 tokens of context is more specific than conditioning the model on 50 tokens of context, and it is natural that the model would estimate the probability of the training data as higher in this situation. However, the result is that some strings are [more] 'hidden' in the model and require more knowledge than others to be extractable.").

197. *See supra* note 72 and accompanying text (discussing the choices model trainers make that can influence the amount of memorization that can be extracted from the model).

kind of existence proof;[198] it shows that an AI system is capable of behaving in this way.

Generative-AI companies attempt to push responsibility for infringement onto users in a variety of ways.[199] Most straightforwardly, they argue that "typical" users do not use generative-AI services to infringe:

> Existing song lyrics are not among the outputs that typical Anthropic users request from Claude. There would be no reason to: song lyrics are available from a slew of freely accessible websites. Normal people would not use one of the world's most powerful and cutting-edge generative AI tools to show them what they could more reliably and quickly access using ubiquitous web browsers.[200]

But this is a fundamentally empirical question. It may be that these adversarial and/or infringing outputs are extremely uncommon, either in absolute terms or as a fraction of the total number of generations made by a system. With the right guardrails in place,[201] it may be the case that extremely few "adversarial" users who try to infringe actually succeed. And perhaps it may be that a generative-AI system, only on extremely rare occasions, produces an infringingly similar output without being explicitly prompted to do so. All of these are testable empirical propositions; they might or might not be true of any specific system at any given time.[202]

---

198. *See supra* Part III.A.2; III.A.3 (discussing the same).

199. *See generally* Widder, Nissenbaum & Grimmelmann, *supra* note 182 (discussing generative-AI providers' deflection of responsibility); Cooper, Moss, Laufer & Nissenbaum, *supra* note 45 (discussing how AI-system builders and deployers evade accountability). *See infra* Part III.I (discussing system-level safeguards).

200. Response at 4, Concord Music Grp., Inc. v. Anthropic PBC, No. 3:23-cv-01092 (M.D. Tenn. Jan. 16, 2024).

201. *See infra* Part III.I.

202. It is also fundamentally a testable, empirical question as to whether a "typical" user would or would not use a generative-AI system in place of a search engine to retrieve information. In the absence of large-scale empirical studies, there is plenty of anecdotal evidence to suggest users rely on generative-AI systems for functionality that they would have previously drawn from search engines. Google integrated Gemini (with retrieval augmented generation, or RAG) into its flagship search product, indicating that the company expects generative-AI chatbots to become an important part of search. OpenAI researchers have cited "overreliance" as a risk of highly capable generative-AI systems. They anticipate user may "excessively trust and depend on the model," which reasonably could include relying on Generative AI to perform more traditional web searches. More broadly, just as it is difficult to separate "adversarial" and "typical" use, it is arguably generally unclear what "typical" use looks like for chatbot systems. *See, e.g.,* Larry Neumeister, *Lawyers submitted bogus case law created by ChatGPT. A judge*

Unfortunately, it is hard to answer most of these questions on the state of present knowledge. The data that would be needed is mostly in the possession of the companies that have developed and deployed these systems. It is possible to make estimates of the fraction of infringing material on YouTube because videos are publicly visible and searchable; it is possible to make estimates of the fraction of infringing views because view counts are also public.[203] But because the typical use case for a generative-AI service is a private generation shared only with the user who requested it, there are no reliable third-party sources of evidence as to how these services are being used in practice. The argument that adversarial uses are uncommon could be right or it could be wrong; we simply do not know, and will not unless and until AI companies share far more information about their usage than they have to date.[204]

Companies also argue that using their services to infringe violates their terms of use, for example:

> Doing so would violate Anthropic's Terms of Service, which prohibit the use of Claude to attempt to elicit content that would infringe third-party intellectual property rights.[205]
>
> We also expect our users to act responsibly; intentionally manipulating our models to regurgitate is not an appropriate use of our technology and is against our terms of use.[206]

With respect, the best analogy for an Internet company discovering that users are violating its terms of service to infringe copyright is Colonel Renault discovering that gambling is taking place in Rick's casino. The Internet is full of pirate sites with *pro forma* disclaimers reminding users not to infringe third parties' copyright. It just so happens that almost everything available through these sites is there without the copyright owners' permission, a fact entirely understood by everyone involved.

---

fined them $5,000, Associated Press, June 22, 2023 (discussing a lawyer using ChatGPT to retrieve case law). *See* Kylie Robison, *Google promised a better search experience — now it's telling us to put glue on our pizza*, The Verge, May 23, 2024, https://www.theverge.com/2024/5/23/24162896/google-ai-overview-hallucinations-glue-in-pizza (detailing issues with Gemini in Google Search). *See* GPT-4 System Card, *supra* note 36, at 19 (defining and discussing overreliance).

203. These estimates may be distorted by various factors, including the difficulty of telling whether an upload is licensed or not, and the fact that many infringing videos are removed.

204. Some models have been released as "open" sets of parameters. This can sometimes lead to more (albeit limited) visibility into how these models are used.

205. Response at 4, *Concord Music Grp., Inc. v. Anthropic PBC*, No. 3:23-cv-01092.

206. OpenAI, *supra* note 188.

More generally, just because behavior is adversarial does not make it atypical. In computer security, robustness is often defined in terms of the adversarial user.[207] Secure systems are expected to be *designed* to resist adversarial usage. A credit-card processor who loses customer financial data to a hacker in a data breach cannot escape responsibility by arguing that the hack was "adversarial" usage. Instead, the expectation is that adversarial users can and will attempt to breach a system and steal or alter data, and it is the responsibility of the system deployer to anticipate and prevent this usage. Similar obligations may or may not be appropriate to impose on the deployers of generative-AI systems. But this is fundamentally a policy question that depends on costs, benefits, incentives, and harms; it cannot be waved away by claiming that "adversarial" usage does not count.

I. Generative-AI System Design

Throughout this Essay, we have predominantly focused on models: models contain memorization and models can be prompted to regurgitate memorized content. But most current copyright infringement lawsuits do not only involve models. They implicate generative-AI *systems* (of which models are just one part), whose construction, deployment, and use embroil an entire, complex supply chain that has important copyright consequences.[208] Generative-AI models are embedded in these systems, which are wrapped in public-facing software services. End users interact directly with these services, not the underlying generative-AI models; interaction with models is indirect, through developer APIs or user interfaces.

Because of this additional surface area, system builders and operators have different places in which they can limit or prevent memorized content in models from being delivered to end users. Even if such content can be regurgitated or extracted from a trained *model*, the additional layers of the *system* can provide insulation that does not expose this content outside of the system. For example:

- On entry, the system can filter or modify user prompts it receives as inputs. Such filters can be other (typically discriminative) machine-learning

207. Indeed, this is an accepted truth in computer-security research, and also grounds definitions of robustness to worst-case scenarios. Nicholas Carlini, Anish Athalye & Nicolas Papernot et al., On Evaluating Adversarial Robustness (2019) (unpublished manuscript), https://arxiv.org/abs/1902.06705 (discussing adversarial robustness in machine learning from first principles). Cooper, Moss, Laufer & Nissenbaum, *supra* note 45 (detailing the relationship between robustness and meaningful notions of accountability for AI/ML systems).

208. *See supra* Part II.B. Talkin', *supra* note 11; Talkin' (Short), *supra* note 42.

models and software that rejects certain user requests before they are ever supplied as prompts to the model.

- The model can be *aligned* in ways that change its response to prompts.[209] For example, to varying degrees of success, alignment can instill behaviors in the model to refuse to produce certain types of content (e.g., memorization).[210]
- For prompts that make it past input filters and are supplied to aligned models, the system can still filter or modify the resulting generations; it can filter the outputs it ultimately delivers to users.[211]

The rhetoric AI companies use to discuss memorization shows that they understand the degree of control they have over their systems. After arguing that the *Times*'s extraction attacks were "not typical or allowed," OpenAI wrote, "we are continually making our systems more resistant to adversarial attacks to regurgitate training data, and have already made much progress in our recent models."[212] These points acknowledge that OpenAI (correctly) anticipates that its systems will be subject to "adversarial attacks" and is designing its systems to make them more "resistant."[213] This admits that planning for and mitigating undesirable user behavior—"adversarial" or otherwise—is a part of doing business when it comes to building and deploying software systems.

At the same time, AI companies also discuss memorization as a kind of "bug"—a deviation from correct system behavior. OpenAI, for example, has

209. *See supra* Part II.B (discussing alignment). Alignment, however, has been shown to be fairly brittle; it is only somewhat effective at resisting undesired user behavior. *See* Nasr, Carlini & Hayase et al., *supra* note 53 (describing breaking alignment in ChatGPT in such a way that surfaces memorization).

210. *See* GPT-4 System Card, *supra* note 36, at 13.

211. Note that discriminative models used in filtering would likely have to be trained on the (copyrighted) data that they would serve to identify for filtering. But discriminative models do not "regurgitate" in the same way that generative ones do; their outputs are not of the same modality as their inputs. *See supra* Part II.A (comparing discriminative and generative models).

212. OpenAI, *supra* note 188.

213. In general, OpenAI has a history of valuing research in adversarial ML and doing "red-teaming" exercises to assess risks. Ian Goodfellow, Nicolas Papernot & Sandy Huang, *Attacking machine learning with adversarial examples* (2017), https://openai.com/research/attacking-machine-learning-with-adversarial-examples (discussing prior research at OpenAI on adversarial ML); OpenAI, *OpenAI Red Teaming Network* (2023), https://openai.com/blog/red-teaming-network (detailing the importance of red-teaming to elicit undesired outputs from models, as a way to assess the risks they present).

written, "'Regurgitation' is a rare bug that we are working to drive to zero,"[214] There are a few things that can be said about this perspective. First, even the rhetoric of "bugs" accepts the reality of regurgitation—that this is a behavior their systems engage in, intended or not. Second, it also accepts that the generative-AI system deployer bears some responsibility for the existence of the bug; it is a known bug in their systems. So, even if we accept that some users are adversarial, they are only capable of being successfully adversarial because there is a known bug for them to exploit in the first place. And third, "feature" and "bug" are essentially contested concepts.[215] As discussed above, memorization may indeed be a feature, not a bug, of learning in large-scale generative-AI models. In some contexts, it may even be a desired behavior, as is the case with memorizing and regurgitating the alphabetized list of 50 U.S. States.[216] It is another question entirely with respect to systems. Even if some undesired memorization is unavoidable for *models*, *system* builders are not off the hook for taking reasonable measures to develop system-level guardrails that prevent surfacing that memorized content to users.[217] In our view, for a company that builds and deploys such systems to argue successfully that memorization reflects internal copying, that copying does in fact have to remain internal to the system.

---

214. OpenAI, *supra* note 188; *see also* Response at 2, Concord Music Grp., Inc. v. Anthropic PBC, No. 3:23-cv-01092 (M.D. Tenn. Jan. 16, 2024) ("Anthropic's generative AI tool is not designed to output copyrighted material, and Anthropic has always had guardrails in place to try to prevent that result. If those measures failed in some instances in the past, that would have been a 'bug,' not a 'feature,' of the product."); *id.* at 7 ("[Claude] is designed to *generate* novel content, not simply regurgitate verbatim the texts from which it learned language. While it does on occasion happen that the model's output may reproduce certain content—particularly texts that escaped deduplication efforts when preparing the training set—as a general matter, *outputting verbatim material portions of training data is an unintended occurrence with generative AI platforms, not a desired result*" (internal citations omitted and emphasis added)).

215. Cooper, Moss, Laufer & Nissenbaum, *supra* note 45 (discussing the porous boundaries between bugs and features in AI/ML: functionally necessary behaviors of AI/ML systems do not always align with social goals); David Gray Widder & Claire Le Goues, What is a "Bug"? On Subjectivity, Epistemic Power, and Implications for Software Research (2024) (unpublished manuscript), https://arxiv.org/abs/2402.08165.

216. *See supra* note 64 and accompanying text.

217. We are not claiming that such guardrails have to be or even can be perfect at fulfilling this goal. Determining what is feasible and reasonable for guardrails involves important empirical and policy questions.

IV. Conclusion: Will the Models be Unbroken?

Nearly four decades ago, computer scientist Allen Newell—a Turing Award winner and AI pioneer—warned legal scholars that they were building their theories about intellectual property and software on a foundation of sand:

> My point is precisely to the contrary. Regardless how the *Benson* case was decided—whether that algorithm or any other was held patentable or not patentable—confusion would have ensued. The confusions that bedevil algorithms and patentability arise from the basic conceptual models that we use to think about algorithms and their use.[218]

His point was not that their policy arguments for and against IP protections were wrong: indeed, he expressed "no opinion" on the patentability of algorithms.[219] Instead, his point was far more fundamental: "The models we have for understanding the entire arena of the patentability of algorithms are inadequate—not just somewhat inadequate, but fundamentally so. They are broken."

Newell's warning has renewed force today. Courts, regulators, and scholars who are grappling with how to apply existing laws to Generative AI—or formulate new ones—must build their theories atop a foundation of conceptual models of how generative-AI systems work, with respect to memorization and much else. If they do not, faulty technical assumptions will lead to ungrounded legal claims—not necessarily wrong, but with no reliable connection to the underlying systems they purport to describe. They need, in short, a good model of models.

218. Allen Newell, *Response: The Models Are Broken; The Models Are Broken*, 47 U. Pitt. L. Rev. 1023, 1023 (1986).

219. *Id.* at 1024.