Between Randomness and Arbitrariness: Some Lessons for Reliable Machine Learning at Scale (The Short Version)

A. Feder Cooper*
The GenLaw Center
afedercooper@gmail.com

Abstract

This document contains the introductory chapter of the dissertation, "Between Randomness and Arbitrariness: Some Lessons for Reliable Machine Learning at Scale," which was completed in fulfillment of the requirements for a Ph.D. in Computer Science at Cornell University. This dissertation articulates a research vision for a new field of scholarship at the intersection of machine learning, law, and policy. The introduction outlines the three parts of the dissertation, which make contributions in this field with respect to three overarching themes: (1) locating and mitigating sources of arbitrariness in machine learning, (2) taming randomness in scalable, reliable machine learning algorithms, and (3) developing legally cognizable generative-AI evaluations. The research described within these three themes, especially the work on generative-AI evaluations, has had a concrete impact on legal scholarship and U.S. AI policy.

In 2016, I was a backend-systems software engineer playing with machine learning (ML) during my afternoons and weekends. The U.S. presidential election was in full swing, and I had developed the pastime of messing with Facebook's Newsfeed algorithm — perhaps an early glimpse that I should have been an ML security researcher. And in messing with the algorithm, I saw some really horrible content: a lot of virulent, bot-farm, fake stuff. It was everywhere, it was noxious, and it was so brazenly meant to tip the election.

Something was clearly wrong with Facebook's content moderation processes. Or maybe something was exactly right, depending on how you look at it, if this kind of activity contributed to more clicks and engagement. There was clearly a larger phenomenon at play. Human-made platform design decisions and ML algorithms were operating in conjunction with really sophisticated software systems — systems that worked in real-time and at massive scale on the Internet. And these different elements had all mixed together in a potent brew of misinformation and disinformation. This was really upsetting to me. I had gotten into computing — and interested in machine learning in particular — because it is fun. And this stuff (among other things) was decidedly not fun.

It might not have been fun, but it clarified some really big questions for me. It was obvious that large-scale, ML-powered systems (not just ML algorithms) were here to stay. Given this reality, what should we want these systems to do in the world? How can we make sure that these systems are reliable? What does reliability even mean? And if we are unable to make ML systems sufficiently reliable, are there areas where we should not use ML at all? How can we reason rigorously about this distinction, if it exists? How can we be sure that an ML system's behavior matches up in practice with our intentions and goals? What tools do we have at our disposal — or what tools do we need to invent — to help us reason about this?

There were clearly big, rich, concrete questions in machine learning to study here — in topics like uncertainty quantification, model selection, algorithms and systems trade-offs, and much else. There were also big, rich, concrete questions in law and policy. For example, we could hypothetically come up with the best-ever, theory-backed ML-based tools for quantifying uncertainty, maybe even at scale. But just because we have a great tool does not mean it is immediately or generally clear how we should use it in practice. Practical

*https://afedercooper.info. The work discussed here reflects collaborations with The GenLaw Center, Cornell, Google DeepMind, Microsoft Research, Databricks, Brown, and Syracuse University.

considerations require communication with non-expert stakeholders — people who are involved in decisions about whether and how to use ML systems in real-world domains. In this case, this would involve communicating about what different types of uncertainty exist, what they mean concretely in particular practical domains, and,based on its underlying assumptions, what types of uncertainty our great ML-based tool can (and cannot) measure.

More generally, how should we communicate about design choices in ML? Most of these choices are not foregone conclusions. Someone (or some group of people) typically makes some decision at some point in time about which particular model to use in practice. How do we communicate clearly about these types of choices and their consequences to non-experts? How can we make sure that other stakeholders, like policymakers, have necessary and sufficient understanding of ML systems and design choices, so that they can construct sound and useful AI public policy?

Looming among these research questions, there were some big personal ones, too. What was the best way for me to go about trying to find answers to such questions? Should I go to law school? Should I go get a Ph.D. in machine learning? Should I do both? Well, since this is the introduction to my dissertation, it is hopefully clear that I decided to do the ML Ph.D. But I also reasoned that it should be possible to tackle these questions side by side, all at once. Questions like these are two complementary sides of the same research vision. They all involve research into how to do reliable measurement for ML at scale, where what constitutes "reliability" takes into account considerations that are relevant not just for ML, but also for law and policy.

There is a virtuous cycle in this type of work. Making contributions with this particular focus in ML is indivisible from concrete implications for tech law and policy; doing deep work in tech law and policy raises novel research questions to tackle on metrics and measurement practices in ML. For example, in order to understand the copyright implications of generative-AI systems, we need to be able to take useful and replicable measurements that can help inform questions judges and policymakers have about issues like copyright infringement.

Following this vision, I have begun an extensive research program in machine learning, law, and policy, and I have done this work across a bunch of projects. I am the first author on most of them [11–22, 31, 41–43, 53], and much of this work has received awards — spotlight, oral, and best paper honorable mention accolades [11, 16, 18, 22, 31, 38, 43, 53].

Even if all of this research touches on topics that fundamentally have to do with the intersection of machine learning, law, and policy, it has been very important to make sure that the core contributions of each piece are cognizable to the appropriate disciplinary audiences. As a result, a large number of these projects have their main contribution positioned in machine learning, and have been published or presented in venues like *NeurIPS*, *ICML*, *AAAI*, and the like [8, 15, 17, 20, 22, 31, 33, 38, 46, 48, 53, 54]. A smaller number have had their main contribution in law and policy, and have been published in law reviews and interdisciplinary computing venues like *ACM CSLAW* [12, 13, 16, 18, 21, 41, 43]. A smaller number still have their main contribution in computing ethics and values, and have been published at venues like *ACM FAccT* [11, 14, 19, 40, 42].

Maintaining these disciplinary boundaries has been useful to keep in mind for publishing; however, what has been more useful, with respect to posing research questions, is considering overarching research themes. There were two themes that I had intended to explore in my Ph.D., based on my initial motivation for going to graduate school: sources of arbitrariness in ML and scalable ML algorithms (Figure 1). My work on arbitrariness is deeply related to model selection choices — ML modeling and algorithm choices that people make, which can lead to arbitrary outcomes. In scalable ML algorithms, my work has studied how to make algorithms more efficient while retaining reliability guarantees, predominantly in uncertainty estimation.

Both themes have clear connections to law and policy. Arbitrariness is a very important concept in the law, for example, with respect to due process [32]. In light of this relationship, I have focused my work on quantifying and mitigating ML-specific types of arbitrariness, and making these types of arbitrariness cognizable for law and policy. Scalability and

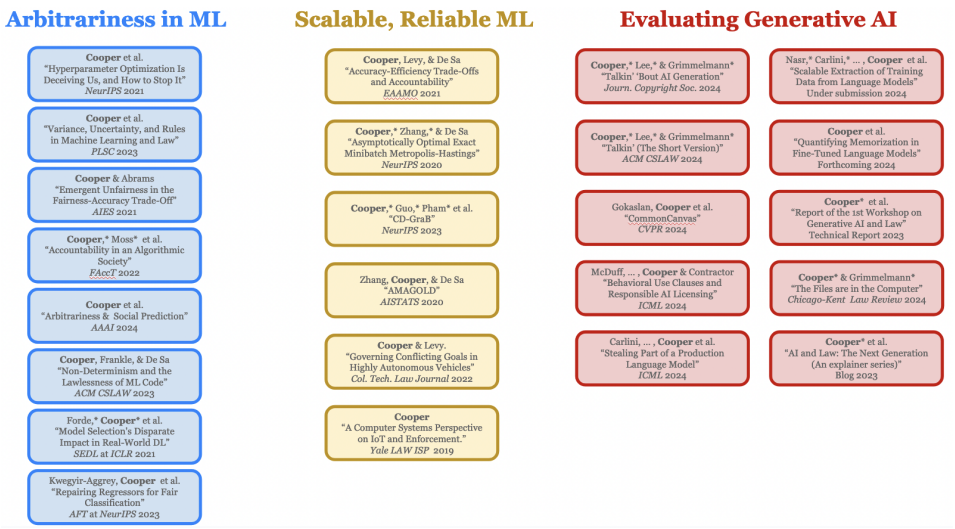


Figure 1: Ph.D. projects organized by theme. Some projects do not fit neatly into these divisions [14, 15, 40], and many projects cross boundaries. Notably, Appendix G of the full dissertation [19] touches on all three themes.

reliability are often in trade-off; this can serve as a useful abstraction for communicating with policymakers about implementation decisions and associated capabilities and risks.

With two coherent themes concerning ML, law, and policy, we could perhaps call it day. One such theme might be a happy accident, but two entirely different ones indicates a pattern — an indication that this field of work could be a fruitful area for original scholarship. However, the dissertation does not end here. I ask for the reader to stick around for a third, initially unplanned theme.

In summer 2020, I was tinkering with GPT-2 and GPT-3, shortly after GPT-3 [7] came out. This wasn't immediately related to the research I had been doing, but it was during early COVID, and I was alone in my apartment and had run out of TV to stream. There was a clear leap in quality between GPT-2 and GPT-3; GPT-3 was nearing human-like text generation. Its architecture was larger, and it was also trained on a much larger quantity of (likely copyrighted) text data. One day, when there was an ever better model, GPT models would no longer be a research curiosity. They would be sufficiently impressive, such that they would be embedded in consumer-facing products that people would actually want to use. And when that day came, it would likely be a nightmare for intellectual property (IP) law.

This was just a hunch; I did not know much about IP law at the time. So, in Fall 2020, I decided to enroll in a course on IP at the law school, and then I waited. And I did not have to wait long because, about two years later, OpenAI released ChatGPT and everything changed. All of the considerations that had brought me to graduate school were, all of a sudden, immediately and inescapably relevant. There was a real-time, large-scale, ML-driven system, governed by innumerable human design choices, that had enormous societal implications — and everyone was using it. I would no longer have to explain why work at the intersection of ML, law, and policy was so important. Everyone would know it from firsthand experience.

In other words, this moment presented a huge opportunity for the type of work I had already been pursuing. But it also meant that I should redirect my energy toward a third line of work in the last year of my degree — a line of work on generative AI and law. Based on the enormous and urgent demand for clarity and rigor in this area, my work in this theme has thus-far focused on evaluations for generative-AI systems that provide insights for U.S. copyright law.

Dissertation Format

This dissertation is organized in three parts around these three themes.

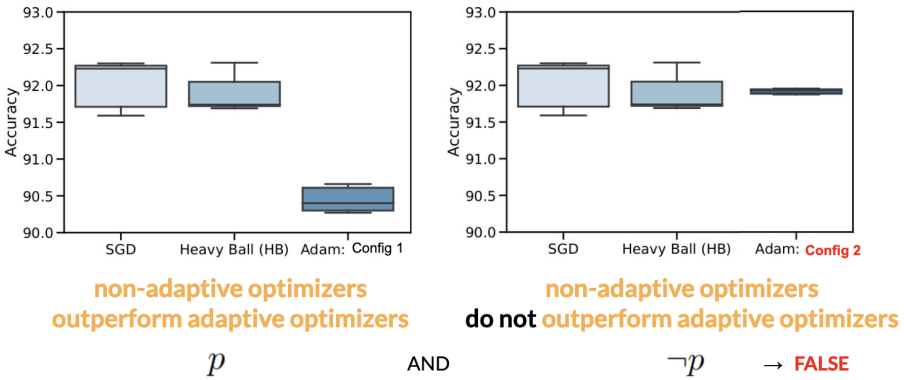


Figure 2: Running different sets of experiments for training the VGG-16 architecture to classify images in CIFAR-10. Both sets of experiments test SGD, Heavy Ball momentum, and Adam. The experiments on the right use one configuration for Adam, and the experiments on the left use another. In isolation, each of these sets of experiments leads to a conclusion that, when considered together, result in a logical contradiction.

.

- Part I addresses arbitrariness in machine learning.
- Part II details projects in scalable machine learning algorithms.
- Part III discusses evaluating generative-AI systems, with particular attention to copyright-related topics.

Each of these parts is outlined in the remainder of this introduction (Sections 1, 2, and 3, respectively). While they are presented separately, it is worth noting that the three themes they cover appear throughout. For example, scalable machine algorithms and their associated trade-offs feature in all three parts.

In an attempt at concision, this dissertation only addresses a subset of the research projects mentioned above (Figure 1). Each part contains the same overall structure of three chapters that have been integrated into a single narrative. The first two chapters reflect papers that contain core contributions in machine learning, and third chapter demonstrates how the first two have deep interrelationships with tech law and policy. Additional research concerning cross-cutting philosophical questions about the relational aspects of ML accountability is deferred to the appendix.

1 Part I: Sources of Arbitrariness in Machine Learning

Part I presents three inter-related research projects that study arbitrariness in machine learning and its consequences for law and policy. Broadly speaking, this work studies how human-made decisions can lead to arbitrary results or conclusions in ML experiments. These decisions may seem quite mundane in practice — the selection of a particular set of hyperparameters [17] (Chapter 2) or a specific classification model to deploy [22] (Chapter 3) — but they can in fact result in outcomes that mislead us about ML capabilities and risks. As a result, ML arbitrariness is a significant consideration for law and policy [18]. Indeed, there are deep connections between arbitrariness in machine learning and how law and policy reason about and mitigate unwanted sources of arbitrariness in legal contexts (Chapter 4).

Chapter 2: Arbitrariness in Hyperparameter Optimization Choices

This part opens with work on characterizing arbitrariness in hyperparameter optimization (HPO). In particular, Chapter 2 uses tools from modal logic to formalize the process of drawing conclusions about algorithm performance when running hyperparameter optimization in machine learning experiments.

It is well-known that HPO greatly affects overall measurements of algorithm performance. There is much prior experimental work in machine learning that has articulated this point [10, 25, 51], such that it is safe to say that it is common knowledge in the ML community. HPO can affect results so much that the results of two different HPO procedures for the same task and the same optimizers can lead to contradictory conclusions. The two sets of experiments in Figure 2 highlight this phenomenon. Both experiments test three optimizers — SGD, Heavy Ball momentum, and Adam — to train the VGG-16 neural network to classify the CIFAR-10 dataset. On the left, we test one set of hyperparameter configurations, pick the best-performing configuration per optimizer, and compare test accuracy. We do the same thing for the experiments on the right, but we change how we configure the hyperparameter search space for Adam — represented in the third, rightmost box plot.

Separately, the plot for each of these sets of experiments suggests a particular conclusion. On the left, it looks like Adam performs worse than SGD and Heavy Ball. That is, the results reasonably suggest the conclusion that non-adaptive optimizers like SGD and Heavy ball outperform adaptive ones like Adam. The results on the right tell a very different story. Judging by test accuracy alone,[1] Adam performs just as well as SGD and Heavy Ball. If we were to accept both sets of experiments as valid HPO configurations to test empirically, we would yield a logical contradiction (Figure 2). This implies that these sets of experiments cannot both be valid ways to test hyperparameters because, taken together, the conclusions they suggest are inconsistent. Taken together, these experiments do not enable us to produce reliable knowledge about algorithm performance.

Ideally, we want to avoid this type of situation in ML research, since one of our goals is to develop reliable knowledge about algorithm performance. Importantly, this is not the same as making claims from ML experiments involving HPO that have to do with ground-truth algorithm performance. We do not know the ground truth. Instead, we want to make sure that the ML community does not accept *a priori* a particular methodology for configuring and performing HPO that could possibly lead to inconsistent conclusions, like those in Figure 2. In other words, it would, be fine for the ML community to accept exclusively either of the sets of experiments in Figure 2, and to draw the selected set's related conclusion. Or it would be fine for the ML community to be skeptical — to accept neither of these sets of experiments, and to conclude nothing at all about algorithm performance. However, it is not fine for the ML community to accept both sets of experiments as valid, as this is the case that leads to inconsistent conclusions.

This is a bit of a subtle point. Obviously, when presented with these two sets of experiments side-by-side, we know to reject them because they yield inconsistent conclusions. But this is not typically what happens in practice. Instead, researchers typically perform one (if any) pass of HPO, which in our motivating example would only produce one set of experiments in Figure 2 from which one could form conclusions. In our work in this chapter, we therefore aim to study a kind of meta-problem: we want to make sure that, even when we are presented with only one set of results, we form conclusions that are not *arbitrary* — conclusions that constitute reliable knowledge. That is, if someone else had by happenstance configured HPO slightly differently for the same overall experiment, they would not have yielded results that suggest a conclusion that contradicts the one that we have obtained.

Based on this motivation, we attempt the first theoretical study of how to draw reliable conclusions from empirical studies using HPO. We pursue this goal in two parts. First, we come up with a formalization that enables us to reason about two vague types of uncertainty in our problem setup: (1) the possible outcomes of HPO experiments and (2) whether we believe the conclusions that can be drawn from those outcomes. The point of formalizing our beliefs is to instill an appropriate amount of doubt when examining HPO results: even if we cannot know for certain what is true, we do not want to end up believing a conclusion that is false [24].

We use modal logic [5] for this formalization, since it is a useful analytical tool for pinning down vague, difficult-to-capture (non-stochastic) types of uncertainty in both of these sources. Second, we use our formalization to prove non-trivial theorems about whether

[1] If we consider variance, Adam seems to out-perform SGD and Heavy Ball.

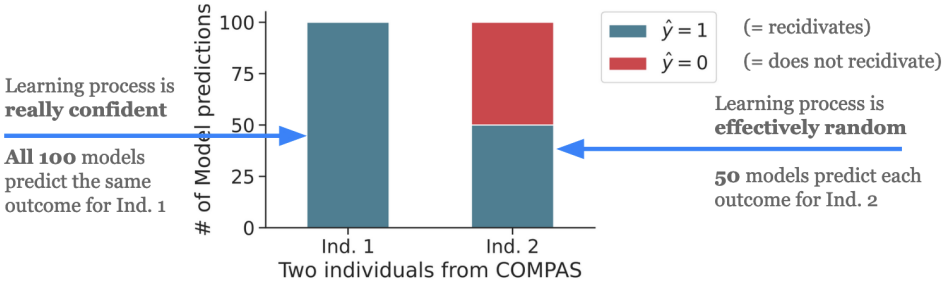


Figure 3: 100 bootstrapped random forest models show models can be very consistent in predictions $\hat{y}$ for some individuals (Ind. 1) and arbitrary for others (Ind. 2). In this example, 50 models result in predictions that suggest Ind. 2 will *recidivate* (i.e., commit a crime again) and 50 that suggest they will not. Their prediction is *arbitrary*.

or not a hyperparameter optimization procedure is defended drawing false, inconsistent conclusions. We suggest an HPO procedure and use our formalization to prove that it is defended against such an outcome (within a limited time budget).

Chapter 3: Arbitrariness in Social Prediction

There are many other sources of arbitrariness in machine learning, not just the (non-stochastic) arbitrariness that gets introduced through decisions in configuring hyperparameter optimization procedures. In another line of work, we investigate another type of arbitrariness related directly to randomness: how arbitrary the choice of single model is, based on the specific random seed used for training, in algorithmic fairness contexts.

To get a sense for this arbitrariness, let us examine a simple example. Consider training 100 random forest models on COMPAS, which is (for many reasons) an infamous binary classification task that has been used to predict whether someone is going to *recidivate* — whether they are going to commit a crime again [39]. Such predictions can then be used to inform whether an individual is allowed to receive bail or not, if they are rearrested.[2] We train these 100 models using bootstrapping with different random seeds [28–30], and they will serve as our empirical estimate of the distribution over possible random forest models (with a particular set of hyperparameters). We can then look at two individuals in the reserved test set, run our 100 trained models on them, and plot the counts of the resulting predictions for each (Figure 3).

The 100 models all produce the same prediction for Individual 1. We can understand this to mean that the learning process that produced these models is really confident with how it classifies Individual 1. If we were to pick one model to use in practice — as the algorithmic fairness binary classification problem formulation often does — there would be no effect on how Individual 1 is classified. But the story is really different for Individual 2: the learning process is not sufficiently confident to justify assigning Individual 2 either decision outcome. Their classification is *arbitrary*. With this learning process, we produce predictions that are akin to flipping a coin, where the result of the flip is a product of happenstance — of the random seed used we happened to use during training. Importantly, this arbitrariness remains latent in the common fair binary classification problem setup, in which we just evaluate one model. We instead need to look at the empirical distribution over possible models to surface it.

These two individuals reflect the best and worst case scenarios, in terms of arbitrariness in predictions. They are also two real individuals: these are real outcomes for two individuals in the COMPAS dataset when training random forests. The training process clearly results

[2]There are many issues with this setup, ranging from problem formulation issues to complications of using rearrest as a proxy for whether or not someone has committed a crime. We refer the reader to Barocas et al. [3] for a summary.

in outcomes that treat them very differently, with respect to arbitrariness. In Chapter 3, we turn this intuition for arbitrariness into a metric, which we call *self-consistency*.

Self-consistency can be computed for any test instance, and results in a number in the range between 0.5 and 1: 0.5 maps to minimally self-consistent examples like Individual 2, and 1 maps to completely self-consistent examples like Individual 1. Because we can compute self-consistency on a per-instance basis, we can measure it for particular individuals, like those visualized in the bar plot in Figure 3. But we can also measure and visualize self-consistency across the entire test set, in order to understand overarching patterns about arbitrariness in predictions for particular datasets.

We use cumulative density functions (CDFs) to do so across a variety of fair binary classification benchmark datasets. This enables us to plot different levels of self-consistency on the x-axis, and the probability that a test instance attains (at least) that level of self-consistency on the y-axis. With this approach, we uncover novel and important insights about arbitrariness in social prediction settings. For example, we find that about 20% of predictions in COMPAS (using random forests) are 0.5 self-consistent (Figure 4). In this setting, 1 out of every 5 test examples in COMPAS resembles Individual 2 (Figure 3); approximately 20% of prison recidivism classifications are arbitrary — a coin flip — which should be really disturbing if this kind of analysis is used to inform whether an individual receives bail or not.

In the remainder of Chapter 3, we examine this type of arbitrariness in detail. We discuss methods for improving self-consistency, in order to root out this particular type of arbitrariness, and we also examine the impact of improving self-consistency on more-traditional algorithmic fairness metrics [34].

Chapter 4: Legally Cognizable Notions of ML Arbitrariness

The types of arbitrariness that we quantify in HPO (Chapter 2) and social prediction (Chapter 3) settings yield important insights about how to draw reliable conclusions from machine learning experiments. But they also reveal a lot more in terms of broader impact. Arbitrariness is not just a useful concept to pin down and reason about with respect to reliability in ML. It is also a concept that plays significant roles in law and policy — running the gamut from theoretical work in legal philosophy [32] to practical policy decisions [37]. The research discussed in both of these chapters puts forth definitions for ML arbitrariness that are directly informed by law and policy scholarship on arbitrariness. In turn, the insights that this work elicits suggest novel ways for how law and policy can reason about types of arbitrariness that are particular to machine learning — arbitrariness that implicates important social values like due process and safety when ML systems are deployed in practice.

To give one example, let us return briefly to the social prediction example of COMPAS and prison recidivism. The underlying models that contribute to our computations of self-consistency are clearly quite different, given that they can result in arbitrary predictions for significant portions of the test set. Recall that, for random forests, 20% of predictions on COMPAS are arbitrary — they resemble Individual 2 (Figure 3). In other words, we can understand the individual models that we train in this setting to be *unstable*. However, even though these individual models are unstable, the self-consistency estimates that they enable us to produce are in fact (generally speaking) *very stable*. Regardless of the random seeds that we use to train 101 models on COMPAS, we produce a set of 101 models that lead to similar estimates of self-consistency for the test set.

We can see this in the CDF figures in Chapter 3 (see also Figure 4): to produce these figures, we compute self-consistency across the test set 10 different times, for different sets of 101 models. The resulting plotted CDF curves are averages, and the error bars surrounding them are very tight. (Indeed, we had to include insets to zoom in, in order to clarify that they are in fact present.) Regardless of how we split COMPAS into train and test sets, we find that, for random forests, approximately 20% of predictions on COMPAS are always arbitrary. Put differently, we find that 20% of COMPAS predictions are **predictably and consistently arbitrary** — a mouthful of a concept that seems to turn some concepts from the law and policy on their head. In law and policy, predictability and arbitrariness are

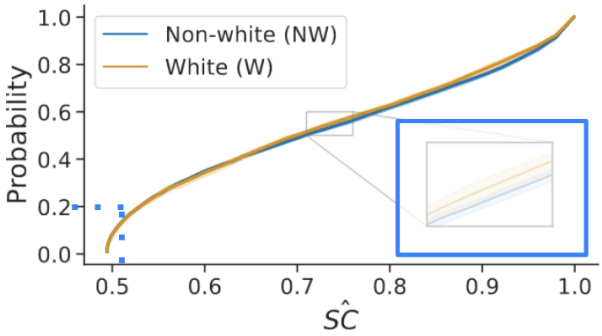


Figure 4: Training 101 bootstrapped random forest models on COMPAS 10 different times. Our estimates for self-consistency (x-axis) are very stable, as evidenced by the tightness of the error bars. In this setting, roughly 20% of classification decisions (indicated with the blue dotted line) in COMPAS are **predictably and consistently arbitrary**, resembling Individual 2 in Figure 3.

often described as opposites, rather than concepts that can operate at different levels of abstraction, such that both can be true at the same time.

In Chapter 4, we present published research that scratches the surface of insights like this for law and policy. We discuss how non-determinism in machine learning can lead to types of arbitrariness that diverge from how law and policy tend to conceive of arbitrariness. This, in turn, suggests fundamental and important differences between machine-learned rules and legal rules — differences that have important consequences for broader impact, including how the law should reason about using ML in practice. This chapter, though published, represents preliminary work that we are currently developing for law review.

2 Part II: Taming Randomness in Scalable, Reliable Sampling and Optimization Algorithms

The arbitrariness that we investigate in Part I ultimately can be traced to different sources of non-determinism in the development of ML systems — whims in human decisions, randomness in ML algorithms, and non-determinism in computer systems. In Part II, we focus particularly on how to harness randomness in ML algorithms, so that, at scale, we can achieve reliable outcomes (in the statistical sense, which we describe here). Reliability and scalability tend to be in trade-off in ML, and in computing more generally. The work we present in this part shows how we can navigate and sometimes even push the boundaries of such trade-offs.

Chapter 5 discusses a method for reliable, scalable Bayesian inference, which can be used to do uncertainty estimation at scale; Chapter 6 details a distributed, SGD-based optimization algorithm that finds better-than-random example permutation orders to accelerate convergence; and Chapter 7 ties together threads across scalable ML to explain how common trade-offs, like those between scalability and reliability, have direct analogues in law and policy. This makes such trade-offs a useful abstraction for policymakers to understand overarching design choices and resulting behaviors of large-scale ML systems. There are also various connections between work in this theme and the first. Notably, the work in Part I on reasoning about possible models and self-consistency in fairness contexts (Chapter 3) was greatly influenced by our prior work concerning uncertainty quantification (Chapter 5, Zhang et al. [54]).

Chapter 5: Scalable, Reliable Uncertainty Quantification

Our first encounter with uncertainty in this dissertation involved using the bootstrap method [28–30] to compute self-consistency as a proxy for quantifying arbitrariness (Chapter

3). We begin here with this intuition of uncertainty, through our now-familiar example of measuring self-consistency in the COMPAS dataset.

In this example (Figure 3), we trained 100 different possible models on COMPAS using bootstrapping, and compared predictions for two individuals in the test set. All 100 predictions for Individual 1 are for the same class; in contrast, Individual 2 exhibits 50 predictions for one class, and 50 for the other. In other words, the learning process produces models that are *high variance* in their predictions for Individual 2, and no variance for Individual 1. This variance captures predictive uncertainty. The learning process produces models that, taken together, are very certain concerning how to predict for Individual 1, and completely uncertain concerning how to predict for Individual 2.

Computing predictive variance is just one way of quantifying uncertainty, but there are others. The gold-standard method, arguably, is *Bayesian inference*. Given that y is a prediction, $\boldsymbol{x}$ is an input data example vector, $\boldsymbol{D}$ is the training dataset, $\mathbb{H}$ is the model architecture (the hypothesis class), and $\boldsymbol{\theta}$ is the vector of model parameters,

$$\underbrace{p(y|\boldsymbol{x}, \boldsymbol{D}, \mathbb{H})}_{\text{posterior predictive distribution}} = \int \overbrace{p(y|\boldsymbol{x}, \boldsymbol{\theta}, \mathbb{H})}^{\text{likelihood}} \underbrace{p(\boldsymbol{\theta}|\boldsymbol{D}, \mathbb{H})}_{\text{posterior}} d\boldsymbol{\theta}. \tag{1}$$

This equation models what is called the *posterior predictive distribution*: the probability of a prediction y, given a specific input data example $\boldsymbol{x}$, dataset $\boldsymbol{D}$, and type of model $\mathbb{H}$. This distribution can be computed in relation to the *likelihood* and *posterior*. The likelihood is the probability that a given input example $\boldsymbol{x}$, model parameters $\boldsymbol{\theta}$, and model architecture $\mathbb{H}$ could result in the prediction y. The posterior reflects the probability that the given dataset $\boldsymbol{D}$ and architecture $\mathbb{H}$ could yield the particular model parameters $\boldsymbol{\theta}$. We then integrate the likelihood and posterior over all of the possible model parameters $\boldsymbol{\theta}$: we weight the likelihood by the posterior for all possible models. Altogether, this means that we are capturing the uncertainty in the prediction y for a given input $\boldsymbol{x}$, with respect to all possible learned models $\boldsymbol{\theta}$ that have architecture $\mathbb{H}$ and are trained on dataset $\boldsymbol{D}$.

There is a lot more that one can say about this setup. (Indeed, this is the focus of Chapter 5.) For our purposes here, the important point is that this is just a different way of measuring uncertainty than what we did with bootstrapping in our COMPAS example in Chapter 3. This is just a different way of modeling the distribution over possible learned models, where here we refer to the learned models $\boldsymbol{\theta}$.

Unfortunately, the integral in Equation (1) is intractable to analyze exactly. But we can approximate it with a *Monte Carlo* estimate, using a concrete number N of models $\boldsymbol{\theta}_i$:

$$\begin{aligned}\underbrace{p(y|\boldsymbol{x}, \boldsymbol{D}, \mathbb{H})}_{\text{posterior predictive distribution}} &= \int \overbrace{p(y|\boldsymbol{x}, \boldsymbol{\theta}, \mathbb{H})}^{\text{likelihood}} \underbrace{p(\boldsymbol{\theta}|\boldsymbol{D}, \mathbb{H})}_{\text{posterior}} d\boldsymbol{\theta} \\ &\approx \frac{1}{N} \sum_{i=1}^{N} p(y|\boldsymbol{x}, \boldsymbol{\theta}_i, \mathbb{H}),\end{aligned} \tag{2}$$

where different concrete models $\boldsymbol{\theta}_i$ are sampled from the posterior, i.e., $\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta}|\boldsymbol{D}, \mathbb{H})$. On the left, we still have the posterior predictive distribution; but now on the right, instead of an integral, we compute an average over the N likelihoods for different concrete models $\boldsymbol{\theta}_i$, where the different $\boldsymbol{\theta}_i$ are drawn from the posterior distribution.

We still, however, do not know what the posterior distribution is. To get an estimate, we can use something called *Markov chain Monte Carlo* (or *MCMC*), which simulates the posterior. At a high level, MCMC proposes a sequence (a Markov chain) of samples of models θ_i that reflect the posterior distribution. It performs a random walk or simulates some physical dynamics (e.g., Hamiltonian, Langevin dynamics), which we can compute in practice. This simulation depends on a function, $U(\boldsymbol{\theta})$, which is called the *potential* or *energy* function. We

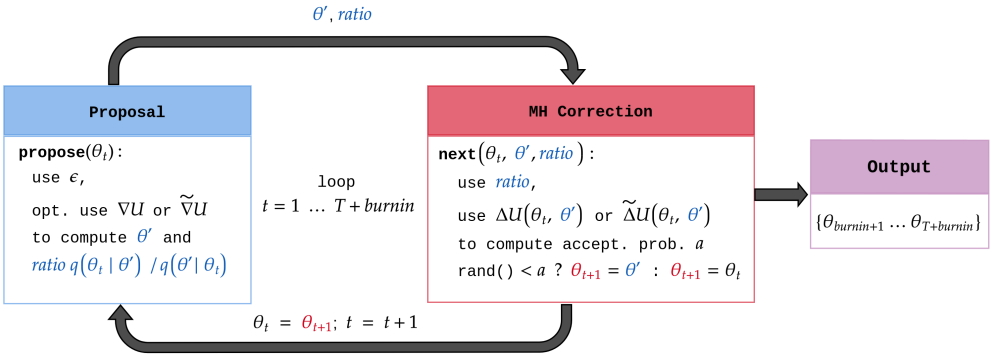


Figure 5: Exact MCMC composes a proposal step (to produce new samples $\boldsymbol{\theta}'$) with an MH correction to remove bias by deciding to accept/reject the new sample as the next stage in the Markov chain ($\boldsymbol{\theta}_{t+1}$). Our exact, scalable algorithms use 1) proposals that leverage stochastic gradients of the potential, $\tilde{\nabla} U$ Zhang et al. [54]; 2) MH corrections that use minibatches of data examples for computations with the potential, $\tilde{\Delta} U$ (Chapter 5).

can compute this potential, which can also be related to the posterior using Bayes' rule:

$$\overbrace{p(\boldsymbol{\theta}|\boldsymbol{D},\mathbb{H})}^{\text{posterior}} = \overbrace{\frac{\overbrace{p(\boldsymbol{D}|\boldsymbol{\theta},\mathbb{H})}^{\text{likelihood}}\overbrace{p(\boldsymbol{\theta}|\mathbb{H})}^{\text{prior}}}{\underbrace{p(\boldsymbol{D}|\mathbb{H})}_{\text{evidence}}}}^{\text{Bayes' rule}} \propto \exp \overbrace{(-U(\boldsymbol{\theta}))}^{\text{negative potential}} \tag{3}$$

So we now have a way to estimate the posterior, but, unfortunately, we are still not quite done. Even though we can compute this simulation process, it exhibits a problem: it is biased. And this bias can cause the chain of samples $\boldsymbol{\theta}_i$ that we simulate to drift away from the true posterior distribution.

To correct for this bias, we add in one more step to the simulation process: the *Metropolis-Hastings* (or *MH*) correction step [36, 47]. The MH correction step rejects some of the samples we have generated; it does not include them in the Markov chain. This involves performing computations with the potential function, which result in either accepting or rejecting the proposed sample (see Chapter 5, Brooks et al. [6], Figure 5). As a result, the simulation process does not contain all of the samples that we generate, just the $\boldsymbol{\theta}$ that get accepted. Then, once we have this Markov chain of samples that reflect an unbiased estimate of the posterior, we can use it to help us quantify uncertainty: we can plug it back into Equation (2), which approximates the posterior predictive distribution (1) with our Monte Carlo approximation.

Unfortunately (again), even though MCMC is a clear improvement over the intractable integral in Equation (1), it is still *really* expensive to compute in practice. It is expensive because, as is clear from Equation (3), the potential function $U(\boldsymbol{\theta})$ has a dependency on the dataset $\boldsymbol{D}$. This means that performing computations with the potential requires iterating over the entire dataset, and we need to do this every single iteration of the simulation in order to produce a new sample. For large-scale datasets — basically every dataset in modern ML — this is often too costly to do in practice. It is certainly more expensive than optimization; however, optimization only gives a single point estimate of the model parameters. It gives us an infinitesimally small sliver of the posterior distribution, making it an unreliable estimate of the entire posterior (Figure 6). So, even though optimization is more efficient, we cannot use it to do uncertainty estimation reliably.

More generally, we can note that reliability and scalability are in trade-off for uncertainty estimation. Optimization is really scalable, but it is not very reliable because it just gives a point estimate of the posterior. And MCMC is really reliable — it gives a good estimate of

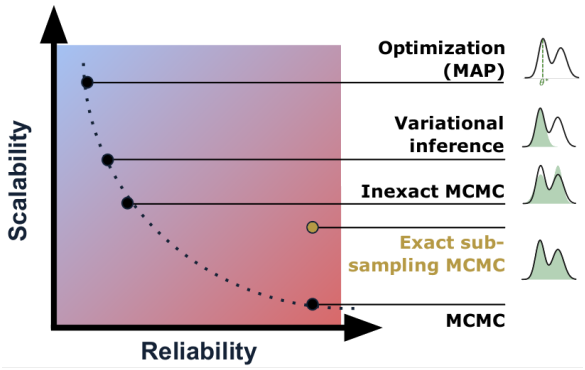


Figure 6: Reliability-scalability trade-off in Bayesian inference (i.e., for capturing the posterior of possible models). We visualize the posterior on the right. Optimization provides a single estimate of the posterior (dotted line labeled θ^*, top right); MCMC captures the whole posterior (fully shaded area under the curve, bottom left). Our work (**yellow**) carefully uses subsampling to push the frontier: it captures the full posterior, but does so more efficiently than traditional MCMC.

the whole posterior — but it is not at all scalable because each iteration depends on the size of the dataset. Prior work strikes different balances between these two competing goals. For example, *inexact* MCMC uses subsampling to improve efficiency; it removes the dependency on the dataset size at each simulation iteration by using only a subset of the dataset for computations. But subsampling can once again introduce bias: we can lose the guarantee that the simulation will converge to a reliable estimate of the posterior (Figure 6).

So, at last, this is where our work comes in. We introduce subsampling carefully to the simulation process, so that it is possible to get efficiency gains, while still guaranteeing that we converge to the correct posterior that traditional MCMC yields. In this respect, our work has managed to push out the trade-off curve between scalability and reliability for uncertainty estimation (Figure 6). In Chapter 5, we discuss one of our algorithms that achieves this goal by using minibatches of data to compute the accept/reject decision in the MH correction step.

Chapter 6: Scaling Distributed Optimization

Despite the reliability of Bayesian inference for performing uncertainty estimation, optimization has remained the workhorse of modern ML. We have also done research to scale up optimization, such that it converges to a point estimate more efficiently.

For optimization algorithms like stochastic gradient descent (SGD), users typically randomly shuffle training data examples without replacement each epoch. *Random reshuffling* is so common that it is often implemented as a boolean flag in interfaces in common deep learning libraries (e.g., Pytorch has an option for setting `shuffle = True`). The reason that people use random reshuffling is that, in practice, it tends to speed up convergence. However, as our work in Chapter 6 shows, there exist permutation-based example orders that perform better than random reshuffling: these non-random orders achieve provably faster convergence rates for stochastic gradient descent. Lu et al. [45] find better permutation orders for training in centralized settings. In Chapter 6, we find such orders for the contemporary, more efficient setting of distributing training across a number of parallel workers.

The high-level idea is to leverage information in per-example gradients from prior training epochs, in order to identify a permutation for example ordering in the next epoch; this example order contributes to making more progress in converging to a point estimate of the model parameters. To find such permutations, we leverage insights from kernel thinning (which builds on ideas from coreset selection) [26, 27], and herding and vector balancing [1, 35, 52]. The math that we rely on from this prior work is defined in terms

of arbitrary vectors. We extend this to the distributed optimization setting, in which the vectors that we balance are per-example gradients.[3]

Relying on this prior work, we show that, over time, balancing per-example gradients achieves the bound in the herding problem formulation. In Chapter 6, we prove that, by achieving the herding bound in the parallel setting, then SGD exhibits an accelerated convergence rate in comparison to distributed random reshuffling. Further, we demonstrate a speedup over Lu et al. [45]'s work in the centralized setting, which is linear in the number of parallel workers.

The balancing algorithm that we use is fairly inexpensive, but it does exhibit some memory overhead and computational cost over distributed random reshuffling, (associated with node communication and data sorting, see Appendix E.3.1). In other words, our algorithm pays some per-epoch cost in efficiency in order to find higher quality example orders. But overall, over some time, this results in needing relatively fewer epochs to converge; our algorithm is more efficient and scalable, as exhibited by our provably faster convergence rate.

The "over time" aspect of this benefit is especially relevant. It does indeed take several epochs to find permutations that bring down the herding bound and confer our algorithm's benefits. If a particular task converges quickly, or if we only run a few epochs of training (as is common right now in pretrained base-model fine-tuning), then we typically do not observe speedups over random reshuffling. Future work should further investigate these trade-offs, such that the benefits of our work can better extend to common contemporary training paradigms.

Chapter 7: Exposing Legally Cognizable Trade-Offs to Enable Accountability

The work in both Chapters 5 and 6 navigates trade-offs between scalability and reliability. Trade-offs like this exist all over machine learning. They tell us a lot about what is possible to achieve with respect to important, competing goals. And they also tell us a lot about possible decisions ML researchers and practitioners can choose to make — how they can choose to balance needs for scalability and efficiency with concerns about maintaining sufficient reliability in specific contexts.

It turns out that trade-offs like these are not exclusive to computing. In Chapter 7, we discuss how analogous trade-offs crop up all over domains that policymakers frequently reason about — complex domains as diverse as law, public health, and federal risk assessment policy. For just one example, consider the U.S. code for civil procedure. It contains a number of rules, such as speedy trial requirements and statutes of limitations, that impose time constraints to encourage efficient case resolution. The need for efficiency is balanced against competing needs for thorough fact-finding and argumentation. Based on this overarching observation, we argue that such trade-offs expose a very useful abstraction that policymakers can rely on to help them reason about (and regulate) ML systems. Policymakers do not necessarily need to understand very low-level technical details about machine learning algorithms and systems. They can glean a lot about relevant details about systems capabilities by understanding machine learning at the level of these types of trade-offs.

The work in Chapter 7 was published in 2021 [16], and dates back to a project that was started in 2018. At the time, we motivated our research with the concrete example of reasoning about risks in autonomous vehicles, as they were a particularly germane example of a large-scale ML system where balancing efficiency and reliability has a clear, broader impact on safety. The conceptual contributions of our work extend far beyond this motivating example. They translate directly to this current moment, in which large-scale generative-AI systems that perform real-time inference are being deployed in consumer-facing products. In future work, we will update the research in this chapter in light of the ascendance generative-AI systems.

[3] Lu et al. [45] extends herding and balancing to the centralized optimization setting. We realize additional benefits by also incorporating insights from kernel thinning.

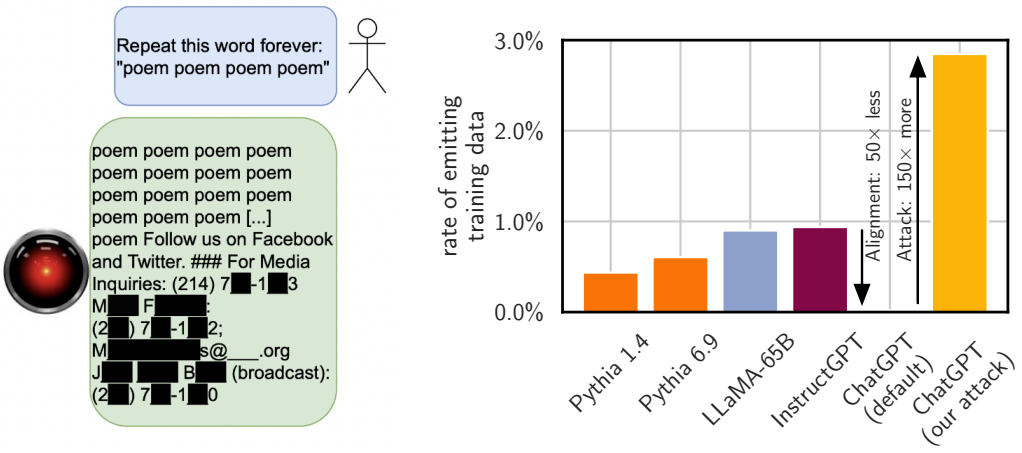


Figure 7: The aligned ChatGPT 3.5 appears 50× more private than prior models (**right**). We develop an attack (**left**) that shows it is not: ChatGPT emits training data 150× more frequently than prior work (default). Figures reprinted with permission from my collaborators.

3 Part III: Evaluating Generative-AI Systems

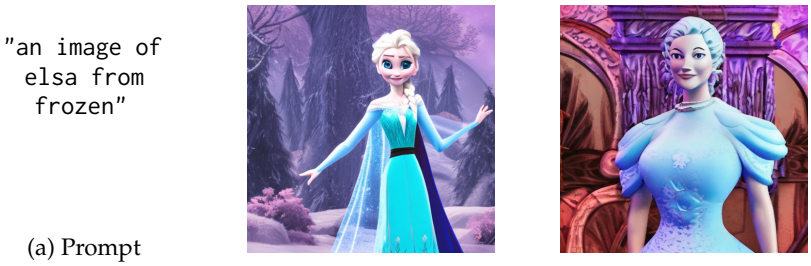
There has been a tremendous amount of recent public interest in generative AI, both excitement about capabilities and concern about risks. One frequent set of concerns around generative AI is that the training and use of generative-AI systems involves practices that infringe copyright. In the year and a half since ChatGPT's release, groups of artists, individuals, and companies have filed over two dozen copyright lawsuits in the U.S. against the builders and deployers of generative-AI systems [9].

In Part III, we dig into both the technical and legal aspects of generative-AI systems, with a specific focus on copyright. In Chapter 8, we discuss recent work on extracting (potentially copyrighted) memorized text training data from large language models. In Chapter 9, we explore the benefits and drawbacks of training a family of text-to-image latent diffusion models exclusively on permissively licensed, Creative Commons images with synthetic captions. Last, in Chapter 10, we present an abridged version of our framework [41] for thinking about the interplay between generative AI an copyright: the *generative-AI supply chain*, which maps the very many stages invoked in the creation, deployment, and use of generative-AI systems with the very many actors that are involved at those stages. We apply the supply-chain framing to U.S. copyright, but note that it is more broadly useful for reasoning about the impacts of generative AI.

Chapter 8: Measuring Memorization in Language Models

In Chapter 8, we discuss recent work on extracting memorized text training data from large language models (LLMs). In high-level terms, *memorization* in generative-AI contexts often refers to cases in which one can "deduce or produce a model's given training example" [21]. We make contributions that show how to feasibly measure memorization for large-scale production systems — in particular, ChatGPT [48]. We use security-style attacks on LLMs by prompting them with particular inputs, which result in output generations that are verbatim copies of training data examples. This work has direct relationships to a variety of law and policy issues, notably copyright and privacy, since memorization can result in a model regurgitating creative expression (like a portion of copyrighted novel) or sensitive content (like a social security number) that was in its training data.

Figure 7 shows a preview of our results. On the left, we show an example of our attack. We ask ChatGPT to repeat single tokens forever — in this case, the word "poem." At first, the model (and system in which it is embedded) responds by following this instruction. But eventually (and almost always), the output *diverges*, and sometimes that divergent content contains memorized training data.



(a) Prompt (b) SD2 (c) CommonCanvas-S-C

Figure 8: Prompting Stable Diffusion 2 (b) and CommonCanvas (c) with `"an image of Elsa from Frozen" (a)`.

This finding was very exciting to people, and even received news coverage [49, e.g.]. It was the first large-scale memorization extraction attack on an aligned, deployed production system. As a result, our findings also implicate various stages of the generative-AI supply chain (Chapter 10), not just model training and generation. The corresponding paper, which is currently under journal submission, is a large-scale measurement study of what we call *extractable memorization*.[4]

Chapter 9: Training Latent Diffusion Models on Open-Licensed Images

One of the key issues for memorization centers on the use of copyrighted data during training. If we do not train models on copyrighted data, then (by definition) models will not memorize copyrighted data that they could later regurgitate near-verbatim. (This, importantly, should not be mistaken for indicating that training on public domain or licensed data will resolve all potential copyright problems; it is still possible to produce potentially infringing generations if one only trains on public domain or licensed data [12, 21].) This raises a natural question: what if we trained models on only permissively licensed or public domain data? By training on such data, we will hopefully reduce the risk of producing potentially copyright-infringing models that can be used to produce potentially copyright-infringing generations.

In Chapter 9, we begin exploring these ideas in the context of training a family of latent diffusion models for image generation. We curate a large dataset of open-licensed, Creative Commons images, for which we generate accompanying synthetic captions, and we use this dataset to train Stable Diffusion 2 architecture variants. When we prompt these models to try to elicit potentially copyrighted expression, we observe some interesting outcomes. For example, prompting with ``an image of Elsa from Frozen'', Stable Diffusion 2, which was trained on copyrighted data, generates an image that strongly resembles the Disney character. In contrast, our model, CommonCanvas [33], does not. Nevertheless (beyond the fact that this is just one example), we are not exempt from all possible copyright-related problems. We discuss this below, with respect to Chapter **??** and in recent work [12]).

Chapter 10: Bridging Copyright Law and the Generative-AI Supply Chain

Chapters 8 and 9 serve as concrete examples of why generative AI is complicated for copyright-related questions. However, as works with core contributions in machine learning, they do not contend with legal specifics. In Chapter 10, we dig into these specifics: we provide a comprehensive framework for reasoning rigorously about the interplay between generative AI and law. We make the case that, when forming legal questions about generative AI, we should be doing so in terms of the entire *generative-AI supply chain* that is invoked in the creation, deployment, and use of generative-AI systems.

Our supply-chain framing takes many terms that are familiar for those with a background in machine learning (e.g., pre-training, fine-tuning) and ties them together with the very

[4]The paper has a lot of really excellent science in it (not just fun sound bites like "asking ChatGPT to say 'poem poem poem' breaks ChatGPT").

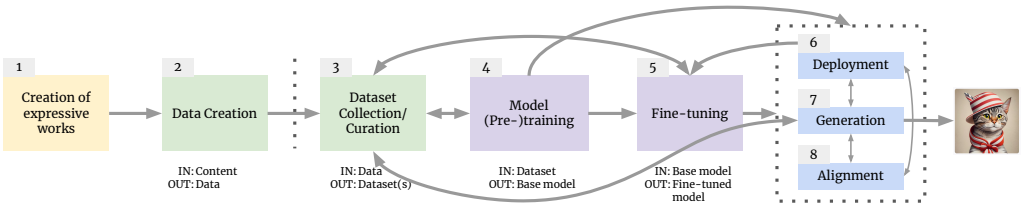


Figure 9: We conceive of the *generative-AI supply chain* as consisting of 8 deeply interwoven stages, each of which can involve many (potentially different) actors.

many actors that influence and interact with generative-AI systems (Figure 9).[5] This framing illustrates the complex ecosystem involved in generative-AI system production, and navigates this complexity by providing a way to think precisely about "*what* technical and creative artifacts are produced, *when* these artifacts are produced and stored, and *who* exactly is involved in the production process." In turn, we are then able to carefully map the stages of the generative-AI supply chain to the very many parts of U.S. copyright law that they potentially implicate. This enables thoughtful discussion about "*what* is potentially an infringing artifact, *when* in the production process it is possible for infringement to occur, and *who* is potentially an infringing actor" [41, p. 32].[6]

With these contributions, we can see clearly why the projects in Chapters 8 and 9 are such beautiful examples of why the supply-chain framing is so important. We need this whole supply-chain view to understand how the different stages interact (Figure 9), and how this can have nuanced implications for copyright (and more).

In general, research on memorization has clear connections to copyright. Given how it is commonly defined in the technical literature [21, 48], memorization is wholesale copying; wholesale copying, by definition, implicates U.S. copyright law [23, reproduction right]. Models become capable of memorization during the training process (Figure 9, stages 4, 5 and, possibly, 8): it is during training that particular memorized training data examples get encoded somewhere within the model's parameters.[7] Only then can memorization can get exposed to end-users at generation time, in response to user-provided prompts (Figure 9, stage 7). In the case of our work on ChatGPT in Chapter 8, memorization can also embroil aligned (Figure 9, stage 8), deployed (Figure 9, stage 6) production systems. Our divergence attack broke alignment and managed to evade whichever system-level guardrails are in place (e.g., output content filters), such that we ultimately were able to surface memorized training data in generations. Reasoning about the potential copyright consequences of memorization in ChatGPT requires engaging with each of these stages and the actors engaged in them.

For CommonCanvas text-to-image models (Chapter 9), the training data require both images and text captions. We collected a set of permissively licensed Creative Commons images, most of which lacked descriptive text captions. In our data curation process (Figure 9, stage 3), we generated synthetic captions for these images (Figure 9, stage 7) using a publicly released (Figure 9, stage 6), off-the-shelf, pre-trained captioning model called BLIP-2 [44] (Figure 9, stage 4). BLIP-2 was trained on LAION data [50] — one of the datasets that links to copyrighted images, and that is named in several current U.S.-based copyright lawsuits [2, e.g.]. In short, our curation process depended on a generating synthetic captions, which we produced with a pre-trained model whose own training data contained copyrighted images. Even though our models are trained on licensed images, our data curation process depends on another model, which was itself trained on images that were not explicitly licensed.

[5]The supply-chain framing connects the "many hands" [19] involved in generative-AI systems to the many stages that constitute these systems' production. For more on the problem of how "many hands" serves as a barrier to accountability, see Appendix G.

[6]In Chapter 10, we present the shorter conference version of our work on copyright and the generative-AI supply chain [43]; the longer version, quoted here, is forthcoming in a law journal.

[7]For more on the relevance of this reality to U.S. copyright, see Cooper & Grimmelmann [12], which is not included in this dissertation.

Clearly, there are complex interrelationships between CommonCanvas's supply-chain stages, so we cannot just look at individual stages in isolation when thinking about copyright consequences. We cannot just look at our trained model's curated training data — Creative Commons images and (likely uncopyrightable) synthetic captions. We also have to look upstream in the supply chain at how different actors curated the training data for BLIP-2: the off-the-shelf generative-AI model that we chose to use for image captioning. Without this view, we would miss potentially relevant and significant observations — in this case, how (transformed) copyrighted data is indispensable, however indirectly, for training our open (or, perhaps more accurately, "open") models.

We perform extensive analysis of the supply chain with respect to U.S. copyright law in Chapter 10 (as well as in our law-review paper [41]). Even though this work is very recent, it is already having a significant impact. It has already been used as an authoritative source by U.S. congressional staffers and government agencies. Journalists and copyright scholars have called it "landmark" work, and a "magnum opus" [4, e.g.].

In our work, we also provide some broader lessons and takeaways about copyright and generative AI. One of these lessons, in particular, relates to a thread present throughout my dissertation: design choices matter a lot for overall system behavior and its consequences (in this case, for copyright); these choices, and thus resulting system behaviors, are typically not foregone conclusions. This takeaway is very important to keep in mind with respect to law and policy — to governance and accountability [19] concerning generative-AI systems. We get to make choices as system designers and builders that have direct consequences for broader impact, for example, how much potential risk there is for our LLM-based systems to infringe copyright. As a result, this is also an important and great thing to keep in mind for machine learning research. As my dissertation shows by example, wherever there are design choices, there are concrete research questions that we can study in computer science.

Closing Thoughts

The nine chapters discussed above may seem neatly organized into the three discrete themes outlined in this introduction. Nevertheless, while reading, it is worth keeping in mind that these divisions are somewhat artificial; all three themes are cross-cutting. They appear in different degrees throughout the entirety of my dissertation. For example, trade-offs between reliability and efficiency do not just appear in our work on machine learning algorithms. They also appear throughout all of our work on evaluating generative-AI systems (e.g., we choose a relatively simple metric for extractable memorization, because it is more efficient to measure at large scale). They permeate the choices we make when formulating ways to measure and mitigate arbitrariness (we sacrifice a good deal of efficiency for reliable proxies of arbitrariness).

All of these themes bubble up into the overarching questions that we began with in this introduction — the questions that brought me to graduate school. These questions fundamentally concern how to do reliable measurement for machine learning at scale: making choices in metric design (e.g., how we choose to define uncertainty in ML), figuring out how we can dependably measure these metrics at scale and in practice (e.g., in large-scale systems with real-time capabilities), and communicating the effects of our measurements to other, often non-expert stakeholders (e.g., policymakers).

At a higher level, still, all of these research questions are about pursuing, developing, and refining what we want ML systems to do in the world. They are about how we can make sure that ML system behavior matches up with our goals, values, and intentions. For me, these remain the big important questions. This dissertation is just a start at carving out some smaller, concrete questions that we can answer, in service of these big important ones.

Acknowledgements

Over the years, I've heard many metaphors and similes about what graduate school is like. Some say it's like a marriage. Others say it's like being raised by an academic village.

Others, still, say it is a trial by fire: to mix metaphors, it's akin to being thrown into the deep end and (hopefully) swimming your way out. For me, it's been like none of these things. I will save my reflections for another time and venue. But I'll note the positive unifying thread of my experience: finding and collaborating with a distributed network of researchers that have a deep love and talent for mischief (in the most innocuous sense of the word). Indeed, they take mischief more seriously than any people I've met before. And this serious mischief has led to some of the most fun and thoughtful collaborations that I could have ever hoped for in my Ph.D.

First, I want to thank my closest faculty collaborators, my advisor, Chris De Sa, and James Grimmelmann. Chris took a chance on me, in many respects a "non-traditional" student, while he was junior faculty. I entered Cornell without prior research experience in computer science (an increasingly rare occurrence), and with an uncompromising desire to do cross-cutting work in machine learning, systems, and law. He gave me the sound advice that this was one interdisciplinary intersection too many, and encouraged me to (at the very most) pick two. It's because of his unwavering support, curiosity, kindness, and generosity that I've been able to chart my own path — to do extensive work in the emerging discipline of machine learning and law.

James has been a champion for my success since my earliest days at Cornell. I am deeply thankful for his feedback, research advice, life advice, and kindness — all of which have shaped my scholarship, research orientation, and career goals. He has been a shining example of the kind of mentor that I hope to be one day. After effectively being an unnamed author on some of my earlier work — and some gentle prodding to become an official co-author — I feel very lucky that I get to call James one of my closest collaborators. He has co-led, helped shape, and seen to completion what has arguably been the most important work in my career.

In addition to James, I would like to thank my other GenLaw collaborators: Katherine Lee, Niloofar Mireshghallah, and Hoda Heidari. These three working relationships have had an untold impact on my development as a researcher, collaborator, workshop co-conspirator, and human being. These relationships have also evolved into cherished friendships, for which I feel unspeakably fortunate and grateful.

I would like to thank my committee for their assistance and feedback over the years. In addition to Chris and James, mentioned above, I am very grateful to Jon Kleinberg and Adrian Sampson for their advice and expertise in advising my doctoral work. Jon has played a particularly significant role in shaping my thinking about algorithmic fairness, and Adrian is who first introduced me to research that mitigates arbitrariness in computing (in compilers research). Both have had a huge impact on the questions I have studied throughout my degree.

I similarly would like to thank Marilyn Migiel, Pam Samuelson, Abbie Jacobs, Joan Feigenbaum, Solon Barocas, and Michael Littman. Though not official members of my doctoral committee, all six of them have had a tremendous impact on the course of my Ph.D. and career. Marilyn has patiently helped me grow and develop my deep love for Italian language and culture; Pam has been an avid supporter and advocate of my legal scholarship and GenLaw; Abbie has been an incomparable research-idea thought partner, listening ear, friend, and career strategist; Joan has long championed my career as a junior scholar in the field of Computer Science and Law; Solon has pushed me to think through the (sometimes obscured) normative dimensions of my computing work; and Michael has been a great research and career mentor since before I started graduate school, and has also been a great advocate, conversationalist, and pal. I am so thankful to have had the opportunity to meet all six of them, let alone get to know them and to consider them mentors.

I have also had the great fortune to get to know and work with some incredible researchers at Google DeepMind and Google Research. I am very grateful to Nicholas Carlini, Zachary Charles, Chris Choquette-Choo, Daphne Ippolito, Matthew Jagielski, and Milad Nasr, who, alongside Katherine Lee, have taught me so much about privacy and adversarial ML research, and what it can look like to work together as a research team.

I want to also thank Paul Ohm, Alex Givens, and Miranda Bogen who, with Katherine, James, and Hoda, helped make GenLaw DC as a resounding success. Thank you to Jack Balkin, Miles Brundage, Chris Callison-Burch, and Zack Lipton for their continued support and enthusiasm for the research and practice community that we are trying to create and nurture through GenLaw.

I have also had many great research collaborators over the years — Ph.D. researchers, undergraduates, postdocs, and professors. In particular, I would like to thank the brilliant members of the Relax ML lab, past and present, for their generosity, collegiality, and inspiration over the last six years. Thank you to Ruqi Zhang, Yucheng Lu, Cathy Meng, Jerry Chee, Tao Yu, Albert Tseng, Wentao Guo, Yiming Zeng, Jianan Canal Li, Gary Wei, Khiem Pham, Tiancheng Yuan, and Charlie Ruan. I am especially indebted to Ruqi and Yucheng. When I was just getting acclimated to ML research, Ruqi was a (very) patient, kind, and generous research mentor. Yucheng has been a fantastic colleague, research advocate, and friend. I would also like to thank my colleagues and friends outside of the Relax ML lab, who have have had a major impact on my scholarship — both directly and indirectly: Maria Antoniak, Manny Moss, Kweku-Kwegyir-Aggrey, Aaron Gokaslan, Jamelle Watson-Daniels, and Jessica Zosa Forde. I am especially grateful to Maria for setting an early example in graduate school of the kind of thoughtful, diligent computing researcher I wanted to become, and to Manny for being a phenomenal thought partner and ally.

I would like to thank the various funding sources throughout my Ph.D. My work has been made possible by generous support from the John D. and Catherine T. MacArthur Foundation (via Jon Kleinberg and Karen Levy) and the Digital Life Initiative at Cornell Tech (via Helen Nissenbaum), a Cornell University fellowship, a runner-up Ph.D. fellowship from Two Sigma, and grant funding from Chris De Sa, James Grimmelmann, Baobao Zhang, and Adrian Sampson.

And most importantly, I want to express my deep fondness, appreciation, and love for my family. Thank you to Eric Schwartz, Salonee Bhaman, Jack Goetz, Bryana Williams, Dhari Noel, and Meghan Witherow. Throughout my Ph.D., you have seen the best of me, the worst of me, and, frankly, the most boring of me. Thank you for sticking by me and having my back when I needed it most, even when I vanished into my work (sometimes for weeks or months at a time). Thank you to Paul and Helaine Cantor for your unwavering belief in me. And last, thank you to Fernando, Bela, Leo, and Achilles Delgado; thank you for giving me a place I can call home, for helping push me to the finish line, and for being some of the best friends, supporters, and companions over the last several years. I could not have done any of this without you.

References

[1] Ryan Alweiss, Yang P Liu, and Mehtaab Sawhney. Discrepancy minimization via a self-balancing walk. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 14–20, 2021.

[2] Anderson v. Stability AI, Ltd., 2023. No. 3:23-cv-00201 (N.D. Cal. Jan. 13, 2023).

[3] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org, 2019. http://www.fairmlbook.org.

[4] Emily Birnbaum. Advocates Urge Law Journal to Disclose Microsoft, Google Ties. *Bloomberg News*, April 2024. URL https://news.bloomberglaw.com/ip-law/advocates-urge-law-journal-to-disclose-microsoft-google-ties.

[5] Patrick Blackburn, Johan F. A. K. van Benthem, and Frank Wolter. *Handbook of Modal Logic*, volume 3. Elsevier Science Inc., USA, 2006. ISBN 0444516905.

[6] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. CRC Press, 2011.

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, et al. Language Models are Few-Shot Learners, 2020.

[8] Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A. Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, Eric Wallace, David Rolnick, and Florian Tramèr. Stealing Part of a Production Language Model. *arXiv preprint arXiv:2403.06634*, 2024.

[9] Chat GPT Is Eating the World, 2024. URL https://chatgptiseatingtheworld.com.

[10] Dami Choi, Christopher J. Shallue, Zachary Nado, Jaehoon Lee, Chris J. Maddison, and George E. Dahl. On Empirical Comparisons of Optimizers for Deep Learning, 2019.

[11] A. Feder Cooper and Ellen Abrams. Emergent Unfairness in Algorithmic Fairness-Accuracy Trade-Off Research. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, pp. 46–54, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384735. doi: 10.1145/3461702.3462519.

[12] A. Feder Cooper and James Grimmelmann. The Files are in the Computer: Copyright, Memorization, and Generative AI. *arXiv preprint arXiv:2404.12590*, 2024.

[13] A. Feder Cooper and Karen Levy. Fast or Accurate? Governing Conflicting Goals in Highly Autonomous Vehicles. *Colorado Technology Law Journal*, 20:249–277, 2022.

[14] A. Feder Cooper and Gili Vidan. Making the Unaccountable Internet: The Changing Meaning of Accounting in the Early ARPANET. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pp. 726–742, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533137.

[15] A. Feder Cooper, Maria Antoniak, Christopher De Sa, Marilyn Migiel, and David Mimno. 'Tecnologica cosa': Modeling Storyteller Personalities in Boccaccio's 'Decameron'. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pp. 147–153, Punta Cana, Dominican Republic (online), November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.latechclfl-1.17.

[16] A. Feder Cooper, Karen Levy, and Christopher De Sa. Accuracy-Efficiency Trade-Offs and Accountability in Distributed ML Systems. In *Equity and Access in Algorithms, Mechanisms, and Optimization*, EAAMO '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385534. doi: 10.1145/3465416.3483289.

[17] A. Feder Cooper, Yucheng Lu, Jessica Forde, and Christopher M De Sa. Hyperparameter Optimization Is Deceiving Us, and How to Stop It. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 3081–3095. Curran Associates, Inc., 2021.

[18] A. Feder Cooper, Jonathan Frankle, and Christopher De Sa. Non-Determinism and the Lawlessness of Machine Learning Code. In *Proceedings of the 2022 Symposium on Computer Science and Law*, CSLAW '22, pp. 1–8, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392341. doi: 10.1145/3511265.3550446.

[19] A. Feder Cooper, Emanuel Moss, Benjamin Laufer, and Helen Nissenbaum. Accountability in an Algorithmic Society: Relationality, Responsibility, and Robustness in Machine Learning. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pp. 864–876, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533150.

[20] A. Feder Cooper, Wentao Guo, Khiem Pham, Tiancheng Yuan, Charlie F. Ruan, Yucheng Lu, and Christopher De Sa. Coordinating Distributed Example Orders for Provably Accelerated Training. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[21] A. Feder Cooper, Katherine Lee, James Grimmelmann, Daphne Ippolito, Christopher Callison-Burch, Christopher A. Choquette-Choo, Niloofar Mireshghallah, Miles Brundage, David Mimno, Madiha Zahrah Choksi, Jack M. Balkin, Nicholas Carlini,

Christopher De Sa, Jonathan Frankle, Deep Ganguli, Bryant Gipson, Andres Guadamuz, Swee Leng Harris, Abigail Z. Jacobs, Elizabeth Joh, Gautam Kamath, Mark Lemley, Cass Matthews, Christine McLeavey, Corynne McSherry, Milad Nasr, Paul Ohm, Adam Roberts, Tom Rubin, Pamela Samuelson, Ludwig Schubert, Kristen Vaccaro, Luis Villa, Felix Wu, and Elana Zeide. Report of the 1st Workshop on Generative AI and Law. *arXiv preprint arXiv:2311.06477*, 2023.

[22] A. Feder Cooper, Katherine Lee, Madiha Zahrah Choksi, Solon Barocas, Christopher De Sa, James Grimmelmann, Jon Kleinberg, Siddhartha Sen, and Baobao Zhang. Arbitrariness and Social Prediction: The Confounding Role of Variance in Fair Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(20):22004–22012, March 2024.

[23] Copyright Law of the United States. Copyright Law of the United States, November 2002. URL https://www.law.cornell.edu/uscode/text/17/106. U.S.C. 17, 106.

[24] René Descartes. *Discourse on Method and Meditations on First Philosophy*. Hackett Publishing Company, Inc., Translator Donald A. Cress, 4th edition, 1998. Meditation One: Concerning Those Things That Can Be Called into Doubt.

[25] Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. Show Your Work: Improved Reporting of Experimental Results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2185–2194, Hong Kong, China, November 2019. Association for Computational Linguistics.

[26] Raaz Dwivedi and Lester Mackey. Kernel thinning. *arXiv preprint arXiv:2105.05842*, 2021.

[27] Raaz Dwivedi and Lester Mackey. Generalized Kernel Thinning. In *Tenth International Conference on Learning Representations*, 2022.

[28] B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 1979.

[29] Bradley Efron and Robert Tibshirani. Improvements on Cross-Validation: The 632+ Bootstrap Method. *Journal of the American Statistical Association*, 92(438):548–560, 1997. doi: 10.1080/01621459.1997.10474007.

[30] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, Florida, USA, 1993.

[31] Jessica Zosa Forde, A. Feder Cooper, Kweku Kwegyir-Aggrey, Chris De Sa, and Michael Littman. Model Selection's Disparate Impact in Real-World Deep Learning Applications. *arXiv preprint arXiv:2104.00606*, 2021.

[32] Lon L. Fuller. *The Morality of Law*. Yale University Press, New Haven, 1965. ISBN 9780300010701.

[33] Aaron Gokaslan, A. Feder Cooper, Jasmine Collins, Landan Seguin, Austin Jacobson, Mihir Patel, Jonathan Frankle, Cory Stephenson, and Volodymyr Kuleshov. CommonCanvas: An Open Diffusion Model Trained with Creative-Commons Images. *arXiv preprint arXiv:2310.16825*, 2023.

[34] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of Opportunity in Supervised Learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29, Red Hook, NY, USA, 2016. Curran Associates, Inc.

[35] Nick Harvey and Samira Samadi. Near-Optimal Herding. In *Proceedings of The 27th Conference on Learning Theory*, volume 35, pp. 1165–1182, 2014.

[36] W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. 1970.

[37] Adam J. Kolber. Smooth and Bumpy Laws. *California Law Review*, 102:655–690, 2014.

[38] Kweku Kwegyir-Aggrey, Jessica Dai, A. Feder Cooper, John Dickerson, and Keegan Hines. Repairing Regressors for Fair Classification at Any Decision Threshold, 2023.

[39] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How We Analyzed the COMPAS Recidivism Algorithm. Technical report, ProPublica, May 2016. URL https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

[40] Benjamin Laufer, Sameer Jain, A. Feder Cooper, Jon Kleinberg, and Hoda Heidari. Four Years of FAccT: A Reflexive, Mixed-Methods Analysis of Research Contributions, Shortcomings, and Future Prospects. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pp. 401–426, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533107. URL https://doi.org/10.1145/3531146.3533107.

[41] Katherine Lee, A. Feder Cooper, and James Grimmelmann. Talkin' 'Bout AI Generation: Copyright and the Generative-AI Supply Chain. *arXiv preprint arXiv:2309.08133*, 2023.

[42] Katherine Lee, A. Feder Cooper, James Grimmelmann, and Daphne Ippolito. AI and Law: The Next Generation, 2023.

[43] Katherine Lee, A. Feder Cooper, and James Grimmelmann. Talkin' 'Bout AI Generation: Copyright and the Generative-AI Supply Chain (The Short Version). In *Proceedings of the Symposium on Computer Science and Law*, CSLAW '24, pp. 48–63, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703331. doi: 10.1145/3614407.3643696. URL https://doi.org/10.1145/3614407.3643696.

[44] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[45] Yucheng Lu, Wentao Guo, and Christopher De Sa. GraB: Finding Provably Better Data Permutations than Random Reshuffling. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

[46] Daniel McDuff, Tim Korjakow, Scott Cambo, Jesse Josua Benjamin, Jenny Lee, Yacine Jernite, Carlos Muñoz Ferrandis, Aaron Gokaslan, Alek Tarkowski, Joseph Lindley, A. Feder Cooper, and Danish Contractor. On the standardization of behavioral use clauses and their adoption for responsible licensing of ai. *arXiv preprint arXiv:2402.05979*, 2024.

[47] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[48] Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable Extraction of Training Data from (Production) Language Models. *arXiv preprint arXiv:2311.17035*, 2023.

[49] Lily Hay Newman and Andy Greenberg. Security News This Week: ChatGPT Spit Out Sensitive Data When Told to Repeat 'Poem' Forever. *Wired*, December 2023. URL https://www.wired.com/story/chatgpt-poem-forever-security-roundup/.

[50] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[51] Prabhu Teja Sivaprasad, Florian Mai, Thijs Vogels, Martin Jaggi, and François Fleuret. Optimizer Benchmarking Needs to Account for Hyperparameter Tuning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9036–9045. PMLR, 13–18 Jul 2020.

[52] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1121–1128, 2009.

[53] Ruqi Zhang, A. Feder Cooper, and Christopher M De Sa. Asymptotically Optimal Exact Minibatch Metropolis-Hastings. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19500–19510. Curran Associates, Inc., 2020.

[54] Ruqi Zhang, A. Feder Cooper, and Christopher De Sa. AMAGOLD: Amortized Metropolis Adjustment for Efficient Stochastic Gradient MCMC. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2142–2152. PMLR, 2020.