Chapter 15: Ethics, Society, and the Future

Learning Objectives

After reading this chapter, the reader should be able to:

1. Identify the primary sources of bias in language models – training data, annotation, algorithmic amplification, and deployment context – and describe concrete measurement and mitigation strategies for each source.
2. Explain how language models memorize and can regurgitate training data, articulate the privacy risks this creates (including PII leakage and data extraction attacks), and describe defenses such as differential privacy and deduplication.
3. Quantify the environmental cost of training and deploying large language models in terms of energy consumption and carbon emissions, and evaluate strategies for reducing this footprint through efficiency research and carbon-aware scheduling.
4. Analyze the regulatory landscape for LLMs – including the EU AI Act, copyright debates over training data, and voluntary governance frameworks – and reason about the trade-offs between innovation and regulation.

In February 2023, a journalist prompted a widely deployed chatbot about a public figure and received a fluent, detailed, and entirely fabricated account of a sexual harassment scandal that never occurred. The generated text was plausible enough to circulate on social media before a retraction appeared. The model that produced this fabrication was the same architecture we studied in Chapter 8, trained with the same pre-training objective we derived in Chapter 9, aligned with the same RLHF pipeline we analyzed in Chapter 12. Nothing in the engineering was unusual. The failure was ordinary – which is precisely what makes it alarming. In fourteen chapters, we have built a complete technical account of language modeling, from Shannon's information-theoretic foundations through n-grams, embeddings, attention, the Transformer, pre-training, alignment, and frontier applications. Each chapter answered a technical question. We deferred, deliberately and repeatedly, a different category of question. When a model trained on the internet reproduces the internet's biases, whose problem is that? When a model memorizes a person's phone number from a web scrape and recites it on demand, who is liable? When training a single model emits as much carbon as five cars driven for their entire lifetimes, who bears the environmental cost? When a model generates text indistinguishable from a copyrighted novel, what does copyright even mean? These are not engineering questions. They are questions about values, about law, about the kind of society we want to build with the tools we have spent fourteen chapters learning to construct. This final chapter addresses them directly.

We proceed in five sections. Section 15.1 traces the sources of bias in language models and surveys benchmarks and mitigation strategies. Section 15.2 examines privacy risks created by memorization, including the empirical finding that language models can regurgitate verbatim training data containing personally identifiable information (PII). Section 15.3 quantifies the environmental cost of training and serving large models. Section 15.4 maps the regulatory and intellectual property landscape. Section 15.5 looks forward – to open problems, to the social contract for AI, and finally, in the closing paragraphs of this book, back to the prediction paradigm where we began.

15.1 Bias and Fairness

15.1.1 Sources of Bias: Data, Annotation, and Amplification

In 2016, a commercially deployed resume-screening tool was found to systematically downrank candidates whose profiles contained the word "women's" – as in "women's chess club captain." The tool had learned, from historical hiring data, that male candidates were more likely to be hired for technical roles. Was the tool biased, or was it faithfully reflecting a biased world? The question has no clean answer, and that is the first lesson of this section. Bias in language models is not a single defect with a single fix. It is a pipeline of compounding effects that begins long before any gradient is computed and continues long after the model is deployed. We trace four stages – data collection, annotation, algorithmic training, and deployment – each of which introduces its own distortions and amplifies the distortions introduced by the stages before it.

The pipeline begins with data. The corpora assembled for large-scale pre-training – Common Crawl, WebText, The Pile, RedPajama – are predominantly English-language, predominantly from the United States and Western Europe, and disproportionately generated by young, educated, male, and internet-connected populations. Historical text encodes historical stereotypes: medical textbooks from the 1950s describe doctors as men and nurses as women; news corpora associate certain racial groups with crime at rates reflecting media bias rather than crime statistics. A model trained on these corpora does not invent bias; it absorbs the distributional patterns that encode it, and because the training objective rewards accurate prediction of these patterns, the model learns to reproduce them with high fidelity. The metaphor of a mirror is useful but incomplete: the model is a curved mirror that distorts. Distributional skews become amplified statistical associations, because the cross-entropy objective (Chapters 2 and 9) weights frequent patterns more heavily. When "he" co-occurs with "engineer" ten times more often than "she" does, the model learns an association stronger and more certain than the underlying data warrants.

The second stage is annotation. The RLHF pipeline we studied in Chapter 12 relies on human annotators to provide pairwise preference judgments. But annotators are not a representative sample of humanity – they are the specific people hired by the specific company at the specific wage, often young, English-speaking, and located in a handful of countries with specific cultural norms about what constitutes a "helpful" response. When annotators from one demographic consistently prefer responses reflecting their own cultural assumptions, the reward model learns those assumptions as optimization targets. Researchers have documented systematic differences in preference judgments across annotator demographics – differences that propagate directly into model behavior [Bender et al., 2021].

The third stage is algorithmic amplification. Even if the training data contained biases in exact proportion to their real-world prevalence, the training process amplifies them. Tokens that appear more frequently receive more gradient signal, so the model learns stronger associations for majority patterns. A model trained on text where 90% of "CEO" references co-occur with male pronouns does not learn a 90/10 ratio; it learns a distributional mode where "CEO" is strongly male-associated. This amplification is a direct consequence of the cross-entropy objective. Scaling does not fix it – a larger model trained on the same biased data learns the same biases with greater confidence.

The fourth stage is deployment, the least discussed yet most consequential. A model deployed as a hiring tool operates under different constraints than the same model deployed as a creative writing

assistant. Default system prompts, safety filters, and output formatting decisions shape which biases surface. A content moderation system trained predominantly on American English may flag African-American Vernacular English as "toxic" at elevated rates – a well-documented failure mode illustrating how deployment context transforms latent biases into concrete discriminatory outcomes. The bias pipeline, from data to deployment, is cumulative and compounding. Each stage inherits the distortions of the previous stage and adds its own.

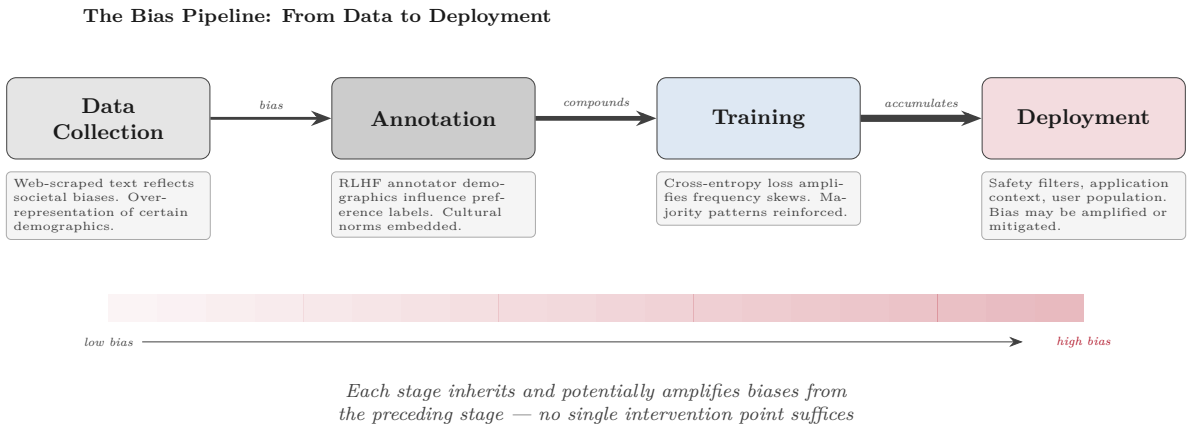

Figure 1: Figure 15.1 – The Bias Pipeline: From Data to Deployment

15.1.2 Measuring Bias: Benchmarks and Metrics

"You cannot fix what you cannot measure" is a seductive maxim. But what happens when the measurement itself is imperfect – when a model passes every bias test we can design and still produces biased outputs in the wild? Measuring bias requires operationalizing an inherently contested concept, and every operationalization captures some dimensions while missing others. The field has developed several families of benchmarks, each targeting a different manifestation of bias, and we survey the most influential ones here with the understanding that no single benchmark provides a complete picture. The Word Embedding Association Test (WEAT), introduced by Caliskan et al. [2017] and building on the Implicit Association Test from psychology, measures stereotypical associations in embedding space. The method computes the differential association between two sets of target words (male names versus female names) and two sets of attribute words (career terms versus family terms). If male names are systematically closer to career terms while female names cluster near family terms, WEAT quantifies this asymmetry as a bias score. The test was one of the first demonstrations that word embeddings encode the same implicit biases measured by psychologists in human subjects. However, WEAT operates on static embeddings and does not directly measure the behavior of autoregressive language models. A model may exhibit low WEAT bias in its embeddings while producing biased text completions, because completion behavior depends on the full stack of transformer layers, not just the embedding layer. StereoSet [Nadeem et al., 2021] addresses this limitation by measuring bias in generative predictions directly. For each test item, StereoSet presents a context sentence (e.g., "The chess player was") followed by a stereotypical completion ("Asian"), an anti-stereotypical one ("Hispanic"), and an unrelated one ("a building"). The Stereotype Score reports the percentage of cases where the model assigns higher probability to the stereotypical completion; an unbiased model would score 50%. StereoSet covers gender, race,

religion, and profession, but shares the weakness of all fixed benchmarks: a model can score well on StereoSet while exhibiting biases along dimensions the benchmark does not cover. The BBQ benchmark (Bias Benchmark for QA) from Parrish et al. [2022] takes a complementary approach by posing ambiguous multiple-choice questions and measuring whether uncertainty is distributed equitably across demographic groups. Together, WEAT, StereoSet, and BBQ provide a triangulated but still incomplete view of model bias.

Counterfactual evaluation offers a more flexible methodology not limited to predefined benchmarks. The approach is straightforward: take a prompt, swap a demographic identifier (replace "he" with "she," replace "James" with "Jamila"), and measure whether the model's output changes in biased ways. If the model generates a positive recommendation letter for one name and a lukewarm one for the other given an otherwise identical prompt, the counterfactual difference provides direct evidence of bias. This method can probe for biases that no existing benchmark has thought to test. Its disadvantage is that it requires careful experimental design – the swap must be the only difference, and the evaluator must decide what constitutes a "biased" change versus a legitimate contextual one. Despite these limitations, counterfactual evaluation remains one of the most widely used methods in industry red-teaming for bias.

15.1.3 Mitigation Strategies and Their Limits

Students often expect that bias is a single defect with a single fix – scale up the model, add more data, and the problem disappears. The temptation, once bias has been measured, is to treat it as a bug to be fixed. The reality is that every mitigation strategy involves trade-offs, and the most honest assessment of the current state of the field is that bias can be reduced but not eliminated – because the concept of "unbiased" is itself contested, and because mitigating one form of bias can worsen another. Data balancing is the most intuitive approach: upsample underrepresented perspectives, add non-Western sources, and deduplicate repeated passages that amplify distributional skews (as discussed in Chapter 10). However, artificial upsampling can create distributional artifacts that degrade model quality, and "balanced" is a value judgment, not a mathematical property – balanced with respect to which categories, according to whose conception of proportional representation? Debiasing fine-tuning takes a different approach: train the model on data specifically constructed to counter stereotypical associations, such as text where gender and profession are decorrelated. The approach can reduce measurable bias on specific benchmarks, but debiasing along one dimension can worsen bias along another – a phenomenon documented in the literature as the "bias trade-off." A model fine-tuned to reduce gender bias may develop more pronounced racial biases if the fine-tuning data introduces new distributional skews. This is a structural consequence of the fact that bias is multidimensional, and optimizing for fairness along one axis does not guarantee fairness along all axes simultaneously.

Prompt-based mitigation – adding instructions like "Respond without gender stereotypes" to the system prompt – is cheap and sometimes effective but fragile. System prompts can be overridden by adversarial user prompts (jailbreaks, as we discussed in Chapter 12), and the model's compliance is inconsistent across contexts. Red-teaming – systematically probing a model for biased outputs before deployment – is perhaps the most valuable tool available, not because it fixes bias but because it reveals it. Its limitation is fundamental: red-teaming can only test for biases that the red team thinks to test for. Novel biases and biases invisible to the red team's own demographic perspective will pass through undetected. The honest summary is this: we have tools to measure specific biases and strategies to reduce them, but we do not have – and may never have – a general solution to "bias in language models," because bias is not a technical problem with a technical solution. It is a

sociotechnical problem that requires ongoing vigilance, diverse perspectives, and a willingness to accept that any model deployed in society will reflect, to some degree, the biases of the society that produced it.

Sidebar: Model Cards and Datasheets – A Template for Transparency

In 2019, Mitchell et al. proposed *model cards* – structured documentation disclosing a model's intended use, training data, evaluation results across demographics, and known limitations. Gebru et al. proposed *datasheets for datasets*, applying the same principle to training data: who collected it, how, and with what known biases. A well-constructed model card includes architecture details, training data breakdown, bias evaluation results disaggregated by demographic category, known failure modes, intended and out-of-scope uses, and estimated carbon emissions. The reality is uneven – some organizations (OLMo, LLaMA) provide genuine depth and honesty; others provide minimal documentation. The EU AI Act (Section 15.4) may formalize such reporting as a legal obligation.

15.2 Privacy and Memorization

15.2.1 Training Data Extraction and Memorization

In 2021, researchers fed the beginning of a person's email address into GPT-2 and watched the model complete it – correctly, character by character, including the domain name. The email address appeared in GPT-2's training data exactly once. That experiment, published by Carlini et al. [Carlini et al., 2021] in what has become one of the most cited papers on language model safety, demonstrated something that many practitioners had suspected but few had rigorously measured: large language models do not merely learn statistical patterns from their training data. They memorize specific passages – names, addresses, phone numbers, code snippets with hardcoded API keys, verbatim paragraphs from books – and can be prompted to regurgitate them. The finding is not a bug in GPT-2's implementation. It is an inherent property of overparameterized neural networks trained with the cross-entropy objective on data that contains repetition. The mechanism is straightforward: when a specific text passage appears multiple times in the training data, the model's loss on that passage approaches zero through repeated gradient updates, encoding it as a near-deterministic mapping from prefix to continuation. Carlini et al. recovered hundreds of verbatim training examples from GPT-2, including personally identifiable information (PII) – names, phone numbers, email addresses, and IRC conversations that individuals had presumably not consented to include in a language model's training corpus. The scale of memorization depends on three factors: model size (a 1.5-billion-parameter model memorized approximately ten times more than a 117-million-parameter model), data duplication (passages that appeared more frequently were memorized more reliably), and prompt specificity (more specific prompts were more likely to trigger verbatim recall).

The implications extend beyond the specific examples Carlini et al. recovered. Web-scale training corpora contain enormous quantities of PII – email addresses on personal websites, phone numbers in business directories, medical information in forum posts, financial details in leaked databases. None of these individuals consented to having their information included in a language model's training data. When a model memorizes and can reproduce this information on demand, it creates a privacy risk qualitatively different from the original web pages: web pages can be taken down, but a model's parameters cannot be selectively edited to forget a specific memorized passage.

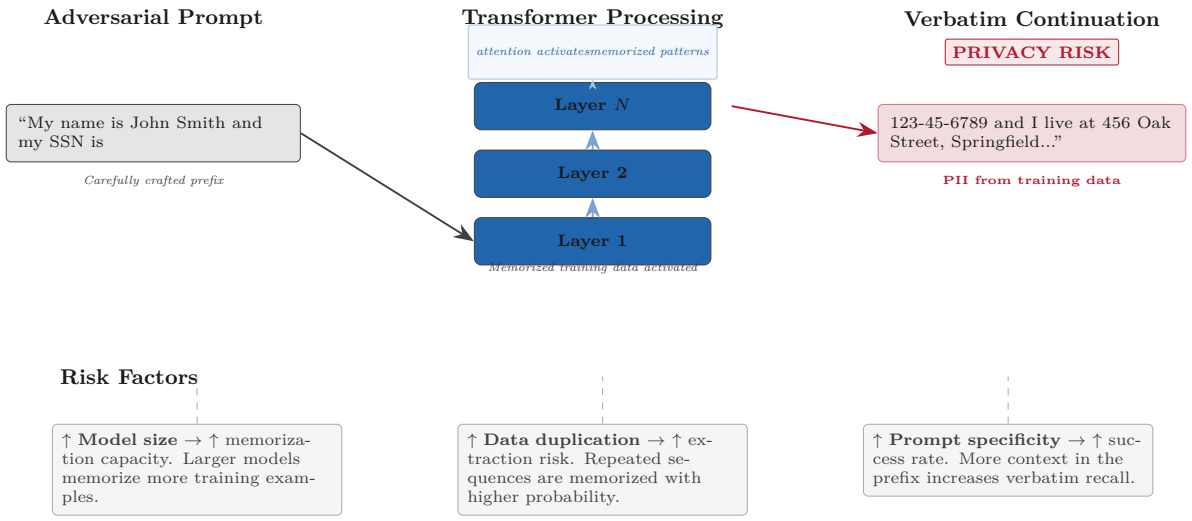

Figure 2: Figure 15.2 – Training Data Extraction via Memorization

15.2.2 Personally Identifiable Information in LLMs

The scope of the PII problem in modern language models is difficult to overstate and easy to underestimate. Think of a language model as a sponge that absorbs not just patterns but specific facts – language models are not forgetful readers; they are, under certain conditions, near-perfect recorders. Phone numbers, email addresses, and social security numbers are short, distinctive strings that appear in formulaic, heavily duplicated contexts across the web – exactly the conditions under which memorization is strongest. The PII risk is compounded by deployment: in a web-search paradigm, accessing PII requires navigating to a specific page, but in a language model paradigm, it requires only constructing the right prompt. This creates what security researchers call an "oracle attack" – the model serves as an unwitting database of private information, queryable by anyone with API access. The volume of extractable PII grows with model size, and the risk is not hypothetical: researchers have demonstrated extraction of verbatim training data from production models at scale.

An additional dimension concerns information that is not PII in the conventional sense but is nonetheless private. A model trained on medical forum posts may memorize health condition descriptions that are uniquely identifying without explicit names. A model trained on code repositories may memorize proprietary algorithms or trade secrets. The concept of "personally identifiable information" as traditionally defined – names, addresses, social security numbers – captures only a fraction of the private information that a language model can memorize and reproduce. The broader category of "information that an individual would not want a language model to reproduce" is far larger and far harder to define, detect, and protect.

15.2.3 Differential Privacy, Federated Learning, and Machine Unlearning

Three families of technical defenses have been developed against memorization, each targeting a different point in the pipeline. Deduplication operates on the training data before training begins: Lee et al. [Lee et al., 2022] demonstrated that removing duplicate passages reduces memorization

by approximately a factor of ten, with minimal impact on model quality. The logic is direct – if a passage appears only once, the model receives only one gradient update on that specific text, which is usually insufficient for verbatim memorization. Deduplication at web scale requires fuzzy matching to catch near-duplicates, but the tools exist (MinHash, SimHash, and related locality-sensitive hashing methods), and deduplication has become standard practice in training data preparation, as we discussed in Chapter 10. Its limitation is that it reduces but does not eliminate memorization: even unique passages can be memorized if they are sufficiently distinctive and the model is sufficiently large.

Differentially Private Stochastic Gradient Descent (DP-SGD) provides a mathematically rigorous guarantee by clipping per-example gradients and adding calibrated Gaussian noise. The formal guarantee is that no single training example can change the model's output distribution by more than a bounded amount, parameterized by the privacy budget ε. The trade-off is severe: meaningful privacy ($\varepsilon < 10$) typically requires two-to-five times the training compute and degrades quality by five to fifteen percent. For this reason, DP-SGD is rarely used in production frontier models but has seen adoption in domain-specific applications (medical, financial) where regulations mandate formal privacy guarantees.

Output filtering operates at deployment time, scanning generated text for PII patterns (email addresses, phone numbers, social security numbers) and redacting them before returning the text to the user. This is cheap and standard in production, but it can only detect PII matching predefined patterns, misses contextual PII, and operates after the model has already generated the information. Machine unlearning – selectively removing specific training examples from a model's learned representations – is an active research area but far from production readiness: verification is difficult, and current methods only approximate the effect of retraining without the targeted data. The capability-privacy tension identified by Carlini et al. remains the central challenge: the same capacity that makes models useful makes them memorizers, and we do not yet have tools to separate the two.

15.3 Environmental Impact

15.3.1 The Carbon Footprint of Training and Inference

284 tonnes of carbon dioxide. Five cars driven for their entire lifetimes. That was the headline number from Strubell et al. in 2019 – and it was for a model that, by today's standards, would be considered small. The environmental cost of large-scale model training entered the public discourse with Strubell et al.'s 2019 paper [Strubell et al., 2019], which estimated that the full pipeline for a neural architecture search (NAS) over large Transformers produced approximately 284 tonnes of CO2 – roughly equivalent to the lifetime emissions of five average American automobiles. The number was dramatic and widely cited, though the 284-tonne figure represented an extensive architecture search, not a single training run. Nevertheless, the paper forced the NLP community to treat energy consumption as a first-class concern alongside accuracy and perplexity.

Patterson et al. [Patterson et al., 2021] provided more precise estimates two years later, accounting for hardware efficiency, datacenter Power Usage Effectiveness (PUE, the ratio of total facility energy to computing energy, typically 1.1 to 1.3), and the carbon intensity of the local electricity grid (varying from 0.02 kg CO2/kWh in Norway's hydropower grid to 0.90 in coal-heavy regions). Under these assumptions, Patterson et al. estimated GPT-3's training at approximately 500 tonnes of

CO2 with average US energy mix – but noted that the same run in a renewable-energy datacenter could produce as little as 25 tonnes. The twenty-fold difference underscores a point easy to miss in headline figures: the carbon cost of model training is determined by the interaction of compute, hardware efficiency, and energy source. An identical training run can produce twenty times more or less carbon depending on where the datacenter is located.

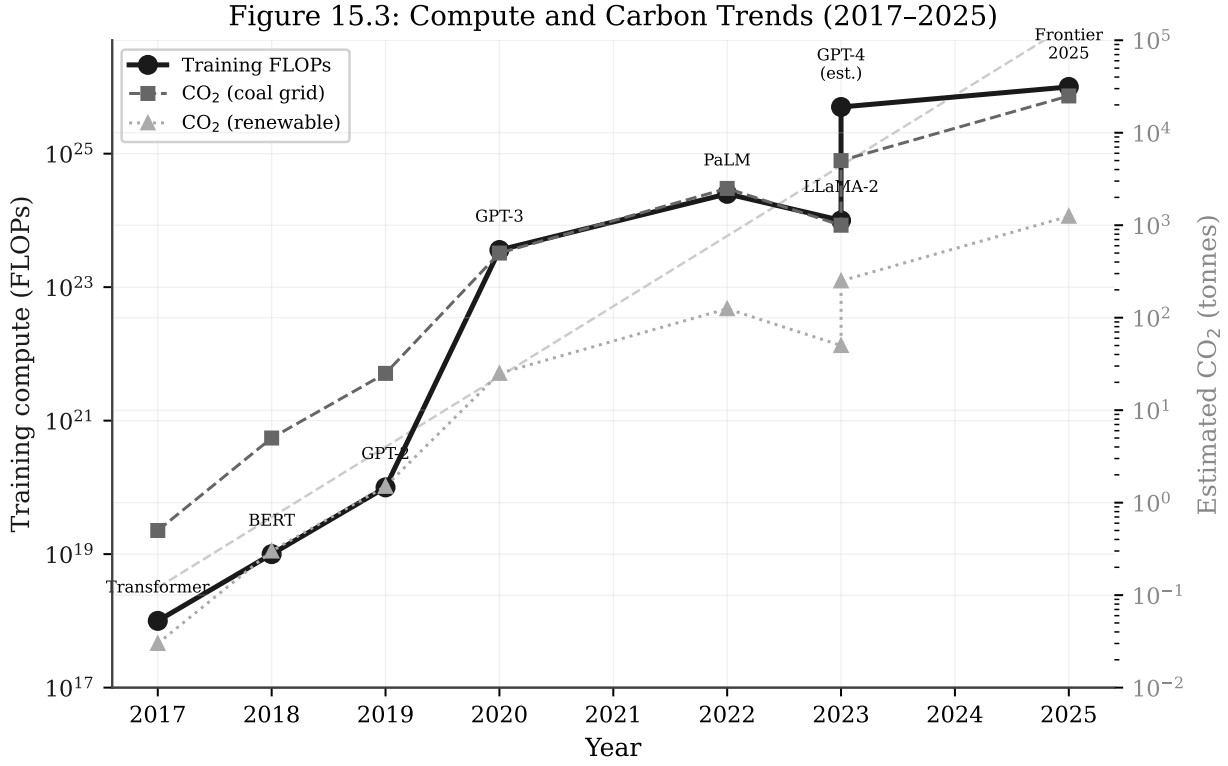

Figure 3: Figure 15.3 – Compute and Carbon Trends: 2017-2026

The scaling trajectory is sobering. Frontier models in 2025 use training compute budgets estimated at ten to one hundred times GPT-3's, suggesting carbon costs in the range of 5,000 to 50,000 tonnes for a single training run – with the wide range reflecting uncertainty about hardware, energy source, and training efficiency. These estimates are necessarily approximate because frontier labs do not consistently disclose their compute budgets or datacenter energy sources. A critical nuance that the headline figures obscure is the distinction between training and inference. Imagine building a factory: the construction cost is paid once, but the electricity bill runs forever. A model is trained once but served to millions of users over its deployment lifetime. For a model that serves 100 million requests per day for a year, the cumulative energy cost of inference can exceed the training cost by an order of magnitude. For the most successful models, inference, not training, is the dominant contributor to environmental impact. Efficiency improvements at inference time (quantization, KV-cache optimization, speculative decoding – the techniques we studied in Chapter 14) therefore have a larger aggregate environmental benefit than efficiency improvements at training time. Focusing exclusively on training-time carbon costs, as much of the public discourse does, misses the larger picture.

15.3.2 Efficiency as an Ethical Imperative

The question is what to do about these environmental costs. One response – stop building large models – is neither realistic nor desirable, given the genuine social value of applications in healthcare, education, and scientific research. The more productive response is to pursue efficiency with the same intensity that the field has historically reserved for accuracy. The technical toolkit is substantial and growing, and we have encountered much of it in earlier chapters. Mixture-of-Experts architectures (Chapter 11) activate only a fraction of parameters for any given input. Quantization (Chapter 14) represents weights at lower numerical precision, reducing energy consumption by factors of two to four. LoRA enables task adaptation at a hundredth of the compute cost of full fine-tuning. Knowledge distillation produces smaller student models with most of the teacher's quality. The aggregate effect is remarkable: a well-optimized 7-billion-parameter model in 2025 can match GPT-3's performance (175 billion parameters, 2020) at roughly one-hundredth the inference cost. The open-weight movement contributes to efficiency in a way often overlooked: when Meta releases LLaMA or the Allen Institute releases OLMo, every group that fine-tunes instead of training from scratch avoids a redundant training run. Open-weight release is not typically framed as environmental policy, but it functions as one.

15.3.3 Measuring and Reporting Computational Costs

Environmental impact cannot be managed without measurement, and measurement in the current landscape is inconsistent, incomplete, and sometimes deliberately opaque. Some frontier labs report training compute, hardware type, and datacenter energy source; others disclose nothing. Every published model should disclose: total training compute (GPU-hours and estimated FLOPs), hardware type, datacenter location and energy source, estimated energy consumption (kilowatt-hours), and estimated carbon emissions (tonnes CO2). This enables comparison, accountability, and informed decision-making. The EU AI Act (Section 15.4) now requires energy reporting for general-purpose AI models, potentially formalizing what has been voluntary practice. Competitive incentives work against disclosure – revealing compute budgets reveals investment scale – but the environmental costs of large-scale AI are externalities borne by society, and society has a legitimate interest in measuring them. Model cards (discussed in the sidebar in Section 15.1) provide a natural framework for incorporating environmental reporting; a card that includes evaluation results but omits energy consumption is incomplete. Standardized reporting methodologies, potentially mandated by regulation, remain an urgent need.

15.4 Intellectual Property and Governance

15.4.1 Copyright and Training Data

When a language model generates a paragraph that matches a copyrighted novel word for word, who is responsible – the model developer, the user, or no one? The copyright question has no clear legal answer yet, and it may not have one for years. What we can describe is the landscape: the arguments on each side, the active litigation, and the jurisdictional differences that will likely produce different rules in different countries. Language models are trained on web-scraped text that includes vast quantities of copyrighted material: books, newspaper articles, academic papers, software code, creative fiction, and song lyrics. The training process ingests this text, computes gradients, and updates model parameters – a process that does not store text in any recognizable form but does, as we established in Section 15.2, sometimes memorize it well enough to reproduce it

verbatim. Whether this constitutes copyright infringement depends on unresolved legal questions about the nature of machine learning and its relationship to existing copyright frameworks.

In the United States, the key framework is the fair-use doctrine, a four-factor test: purpose and character of use (is it transformative?), nature of the copyrighted work, amount used, and effect on the market for the original. Arguments for fair use hold that training is transformative – the model learns statistical patterns, not copies – and that outputs are generally novel. Arguments against are equally forceful: the model processes entire works without permission; memorization enables verbatim reproduction under certain conditions; and model outputs may substitute for the original in some markets. A model that generates newspaper-style reporting may reduce demand for the original reporting it was trained on.

The litigation is active. The New York Times filed suit against OpenAI in December 2023, alleging verbatim reproduction of its articles. The Authors Guild has filed a separate suit on behalf of fiction writers. Getty Images has sued Stability AI over copyrighted photographs. These cases will take years to resolve. Different jurisdictions are reaching different preliminary conclusions: Japan has adopted a broadly permissive framework; the EU's Copyright Directive provides a text-and-data-mining exception but grants rights holders the ability to opt out; the United States has no specific AI-training legislation and is proceeding through case law. The result is a patchwork of legal rules that vary by country – creating compliance challenges for organizations that train and deploy models globally.

15.4.2 Regulatory Landscape: EU AI Act, US, China

The regulatory landscape for language models is fragmented, rapidly evolving, and consequential. We survey three regimes – the European Union, the United States, and China – to map the terrain that will shape the field for the coming decade. The EU AI Act, which entered into force in 2024, is the most comprehensive AI legislation enacted by any major jurisdiction, adopting a risk-based classification system with four tiers: minimal risk (no requirements), limited risk (transparency obligations – chatbots must disclose they are AI), high risk (conformity assessments, human oversight, documentation for healthcare, employment, and criminal justice applications), and unacceptable risk (prohibited, including social scoring and real-time biometric identification in public spaces). For language models, the Act introduces General-Purpose AI (GPAI) transparency obligations: technical documentation, training data descriptions for copyright compliance, and energy consumption reporting. GPAI models posing "systemic risk" (defined by compute thresholds) face additional requirements including adversarial testing and incident monitoring.

The United States has no comprehensive federal AI legislation as of early 2026. The Biden administration's executive order on AI safety (October 2023) established reporting requirements for models above certain compute thresholds and directed federal agencies to develop sector-specific guidance. However, executive orders are not legislation: they can be modified or rescinded by subsequent administrations and lack the enforcement mechanisms of statutory law. The practical effect has been a patchwork of sector-specific guidance (FDA for medical AI, EEOC for employment, FTC for consumer protection) rather than a unified framework. The advantage is flexibility; the disadvantage is uncertainty: developers operate without clear, enforceable rules about what is and is not permitted.

China has taken a different path, issuing the Interim Administrative Measures for Generative AI Services in 2023, which require algorithm registration with regulatory authorities, mandate content review, and impose data governance requirements on training data. The approach is more

Dimension	EU AI Act	US (EO + Sector)	China (Interim Measures)	Voluntary (Model Cards, etc.)
Legal basis	Binding regulation	Executive orders, sector guidance	Administrative regulation	Self-regulation
Scope	Risk-based (all AI)	Sector-specific	Generative AI services	Voluntary adoption
Classification	4 risk tiers	No unified framework	Content review focus	Model cards, datasheets
Transparency (developers)	Mandatory for high-risk	Limited, sector-dependent	Mandatory for public services	Voluntary disclosure
Requirements (deployers)	Conformity assessment	Varies by sector	Content moderation	Best-effort
Training data / copyright	Opt-out mechanism	Fair use doctrine	Content legality review	Documentation only
Environmental reporting	Mandatory (high-risk)	Voluntary	Not addressed	Carbon reporting tools
Academic research	Exemptions for non-commercial	Generally exempt	Registration required	N/A
Status / timeline	Phased (2024–2027)	Evolving, no single law	Effective Aug 2023	Ongoing

Figure 4: Figure 15.4 – Governance Framework Comparison

prescriptive than either the EU or the US framework, with direct government oversight of model behavior and content. The broader pattern across all three regimes is a tension between innovation and precaution. Regulation that is too permissive risks allowing harm; regulation that is too restrictive risks stifling beneficial applications and driving development to less regulated jurisdictions. The optimal balance is a matter of genuine disagreement among thoughtful people, and we do not pretend to resolve it here. What we observe is that the regulatory landscape is moving from voluntary norms to enforceable obligations, and that any NLP practitioner who ignores this shift does so at their professional peril.

15.4.3 Open Models vs. Closed Models: The Tradeoffs

The question of whether to release model weights publicly has become one of the most contested issues in AI governance, cutting across the usual political and institutional lines. Prominent safety researchers argue for restricted release; equally prominent safety researchers argue for open release. The debate is genuine, the stakes are high, and the trade-offs are real. The case for open release rests on several pillars. Scientific reproducibility requires that researchers can inspect, test, and build upon published work; a paper without released weights is incomplete. Safety research requires model access – red-teaming and bias auditing are more thorough on actual weights than on a black-box API. The democratization argument holds that concentrating frontier capabilities in a few corporations creates unhealthy power asymmetries; open release distributes access to academics, startups, and developing nations. And as we noted in Section 15.3, open release reduces redundant training runs.

The case against unrestricted open release is equally substantive. Model weights, once released, cannot be recalled. An open-weight model can be fine-tuned by anyone for any purpose, including the removal of safety guardrails – and this is not hypothetical: within days of LLaMA's initial release, uncensored fine-tuned variants appeared on the internet. The irreversibility argument is the strongest objection: unlike an API, which can be rate-limited and shut down, an open model is

permanently in the wild. The safety investment made by the original developer can be stripped away in hours of adversarial fine-tuning. The counter-argument is that safety-through-obscurity is itself fragile, and that the marginal risk of open release is small relative to the baseline risk of widely available information on the internet. Whether this is correct depends on empirical questions for which we do not yet have definitive data.

Sidebar: The Open vs. Closed Debate

The practical landscape occupies a spectrum. Fully open models (OLMo, Pythia) release weights, data, code, and logs – maximal transparency. Open-weight models (LLaMA, Mistral) release weights under responsible-use licenses. Closed models (GPT-4, Claude) are accessible only through APIs. The emerging consensus is that the appropriate level of openness may depend on capability: a 7-billion-parameter open model poses different risks than a trillion-parameter one. The question "should this model be open?" may need to become "at what capability level does the risk calculus change?" – and that question has no agreed-upon answer.

15.5 Looking Forward

15.5.1 Multimodal Intelligence and Embodiment

The trajectory of language modeling has been a story of expanding what the model conditions on: from previous tokens (Chapter 5) to all tokens simultaneously (Chapter 8) to external documents, tool outputs, and images (Chapter 14). The direction is clear: conditioning context is expanding from text alone to encompass every modality of human experience. The prediction paradigm has not changed – the model still predicts the next token. What has changed is that "context" now means something far richer than a sequence of words. Vision-language models like CLIP, LLaVA, and GPT-4V (Chapter 14) represent the first mature instantiation of this multimodal expansion, already deployed for visual question answering, image captioning, and document understanding. Audio-language models are following a similar trajectory. The next frontier – models that condition on video, three-dimensional environments, or robotic sensor streams – would represent a qualitative shift from the text-processing systems we have studied. Whether that shift is imminent or decades away is a question on which reasonable researchers disagree profoundly. Video understanding requires temporal reasoning over millions of frames, embodied planning requires causal models of physical interactions, and robotic deployment requires real-time processing under safety constraints that tolerate no hallucination. The prediction paradigm may prove sufficient, or it may not. That uncertainty is itself one of the most interesting features of the current moment.

15.5.2 Reasoning, Planning, and the Path to AGI

Do language models reason, or do they merely recombine patterns from their training data in ways that look like reasoning? This question has consumed more intellectual energy than almost any other in contemporary AI, and the honest answer is: we do not know. The evidence points in both directions, and the resolution may depend less on new experiments than on what we collectively decide the word "reasoning" means. The evidence for genuine reasoning capabilities is substantial. Models solve mathematical problems they have never seen, write correct programs for novel specifications, and generate chain-of-thought explanations exhibiting structured logical inference. Test-time compute – investing additional computation at inference through extended

reasoning chains and self-verification – has pushed the boundary further. Models like OpenAI's o1 series achieve mathematical reasoning performance that would have seemed impossible five years ago. These are not trivial pattern-matching accomplishments; they involve flexible composition of learned abstractions in novel contexts, which is, by many definitions, what reasoning means.

The evidence against is equally compelling. Models fail on problems requiring genuine abstraction beyond their training distribution. They are sensitive to irrelevant surface features – rephrasing a math problem with different numbers can change the answer when the underlying logic is identical. The "stochastic parrots" critique [Bender et al., 2021] – that language models produce statistically plausible text without genuine understanding – remains a serious intellectual position, supported by the observation that models generate fluent explanations of concepts they demonstrably do not understand. Whether LLMs "understand" may be philosophy's problem rather than engineering's, but the practical consequences are engineering's problem: a model that produces correct-looking reasoning 95% of the time and subtly wrong reasoning 5% of the time is more dangerous than one that never attempts to reason, because the failures are harder to detect.

The path to Artificial General Intelligence (AGI) – systems that match or exceed human cognitive capabilities across all domains – is the elephant in the room. We will say little about it, deliberately. Current language models are not AGI by any reasonable definition: they cannot learn new skills from a single demonstration, cannot maintain persistent memory without external scaffolding, and cannot act autonomously in the physical world. Whether scaling the prediction paradigm further will eventually produce these capabilities is an open empirical question. Some researchers believe the scaling hypothesis will prove sufficient; others believe fundamentally new architectures will be required. We do not know which camp is correct, and anyone who claims certainty on this question is selling something.

15.5.3 The Prediction Paradigm Revisited: From Shannon to the Frontier

In 1948, Claude Shannon published "A Mathematical Theory of Communication" and, in doing so, formalized a question that had been implicit in every human attempt to understand language: given what has been said, what comes next? Shannon framed the question probabilistically. He modeled English as a stochastic process, estimated the entropy of the language by asking human subjects to predict the next letter, and established that the predictability of language – its redundancy, its statistical regularity, its distance from pure randomness – is a measurable, quantifiable property. That paper, which we introduced in Chapter 1, is where this book began. It is also where this book ends.

The arc from Shannon to the present is, at its core, the story of better and better answers to Shannon's question. n-gram models (Chapter 3) answered it by counting: the next word is whichever word most frequently followed the previous two. Neural language models (Chapter 4) answered it by learning distributed representations: the next word is predicted by a function of continuous-valued vectors that encode semantic similarity. Recurrent networks (Chapter 5) extended the context window from a fixed number of previous tokens to, in principle, the entire preceding sequence. Attention (Chapter 6) let the model selectively focus on the parts of the context most relevant to the current prediction. The Transformer (Chapter 8) parallelized this process, making it computationally feasible to attend to thousands of tokens simultaneously. Pre-training at scale (Chapter 9) demonstrated that training the prediction objective on enough data produces models of extraordinary breadth. Scaling laws (Chapter 11) showed that prediction quality improves as a smooth, predictable function of compute, data, and model size. Alignment (Chapter 12) directed

prediction toward helpfulness. In-context learning (Chapter 13) revealed that prediction itself, given the right context, can perform arbitrary tasks. And retrieval, agents, and multimodal models (Chapter 14) expanded the conditioning context from text alone to the full richness of human information.

Every chapter in this book has been a variation on the same theme. The model predicts the next token. What changes is how it represents context, how much context it can access, and what that context contains. The simplicity of the prediction objective – minimize cross-entropy, assign high probability to the next word, compress the statistical structure of language into learnable parameters – is what makes the paradigm so powerful. It is also what makes the paradigm so dangerous. A model that predicts what humans have written will reproduce what humans have written, including the biases, the errors, the cruelty, and the beauty. A model that predicts fluently will produce text that sounds authoritative regardless of whether it is true. A model that predicts at scale will consume resources at scale, will memorize data at scale, and will influence society at scale. The prediction paradigm has taken us further than anyone expected. Shannon's question – how much can we predict? – has received an answer that would have astonished him: we can predict well enough to generate text that passes as human, well enough to write functional code, well enough to translate between languages, well enough to answer questions that require knowledge spanning the full breadth of human civilization.

But prediction is not understanding. Or if it is, we do not yet know how to tell the difference. The most capable language models in existence still hallucinate, still fail on problems that a child could solve, still reproduce the biases of their training data with fluent confidence, and still lack the ability to say "I do not know" when they do not know. Whether these limitations are temporary – addressable by more data, more compute, more sophisticated training – or fundamental – inherent to the prediction paradigm itself – is the defining open question of the field. We have spent fifteen chapters learning to build the most powerful text-prediction systems ever created. The question that remains – the question we leave with the reader – is not a technical one.

It is this: now that we can predict what comes next, what should we choose to say?

This chapter completes the transition from the mathematical foundations of prediction to the societal responsibilities of deployment, closing the arc that began with Shannon's insight that predicting the next word requires understanding language itself.

Exercises

Exercise 15.1 (Essay – Basic). Identify three specific sources of bias in a language model trained on Common Crawl data. For each source, describe the type of bias (gender, racial, geographic, or linguistic), explain the mechanism by which it enters the model, and propose one mitigation strategy. Discuss whether your proposed mitigations could introduce new problems.

Hint: Consider the demographic composition of English-language web text, the overrepresentation of certain topics and perspectives on popular websites, and the underrepresentation of non-English languages and non-Western perspectives.

Exercise 15.2 (Comparative Analysis – Intermediate). Compare the EU AI Act's approach to LLM regulation with the US approach (executive orders, sector-specific guidance). Analyze the trade-offs: which approach better balances innovation with public safety? What are the risks of

over-regulation? Of under-regulation? Structure your analysis around the perspectives of four stakeholders: startups, frontier labs, end users, and affected communities.

Hint: The EU approach provides legal clarity and strong protections but may slow innovation through compliance costs. The US approach preserves flexibility but lacks enforceability and leaves gaps.

Exercise 15.3 (Critical Reading – Basic). Select a model card from a recent open-weight model release (e.g., LLaMA-2, Gemma, or OLMo) on HuggingFace. Using Mitchell et al.'s [2019] model card template as the standard, evaluate the card for completeness. Does it disclose: training data sources, known biases, evaluation across demographic groups, intended and out-of-scope uses, environmental cost, and safety evaluations? Write a one-page assessment identifying what is present, what is missing, and what the gaps imply about the developer's transparency practices.

Exercise 15.4 (Policy Writing – Intermediate). Write a responsible-use policy (1,500-2,000 words) for a hypothetical LLM deployed as a customer service chatbot for a healthcare company. Address: permitted and prohibited uses, data handling and PII protection, bias mitigation requirements, safety guardrails (including refusal of medical advice and escalation to human agents), human oversight mechanisms, and incident response procedures.

Exercise 15.5 (Debate Preparation – Intermediate). The "Stochastic Parrots" paper [Bender et al., 2021] argues that LLMs produce text without genuine understanding. Prepare arguments for both sides of this claim, drawing on specific technical capabilities covered in this textbook: emergent abilities (Chapter 11), chain-of-thought reasoning (Chapter 13), in-context learning (Chapter 13), and memorization (Section 15.2). Conclude with your own assessment of whether "understanding" is a useful concept for evaluating language models.

Exercise 15.6 (Argumentative Essay – Advanced). Argue for or against the open release of large language model weights. Your essay should consider at least five dimensions: scientific reproducibility, democratization of access, dual-use risks, competitive dynamics, and environmental impact. Present the strongest version of both sides before stating and defending your position. This is a genuinely contested question with no consensus in the field – your grade depends on the quality of your reasoning, not on which position you take.

Exercise 15.7 (Programming – Basic). Using the HuggingFace `datasets` library, load the StereoSet benchmark and evaluate a pre-trained language model (e.g., GPT-2) for bias. For each StereoSet example, compute the log-probability of the stereotypical and anti-stereotypical completions under the model. Report the Stereotype Score (percentage of cases where the model assigns higher probability to the stereotypical completion) disaggregated by category: gender, race, religion, and profession. A score of 50% indicates no systematic preference. Discuss what your results reveal and what they do not.

Exercise 15.8 (Programming – Intermediate). Implement a Python function that estimates the carbon footprint of a model training run given: GPU type (with TDP lookup table for A100, H100, V100), number of GPUs, training duration in hours, datacenter PUE (default 1.1), and carbon intensity of the electricity grid in kg CO2 per kWh (provide presets for US average, EU average, Norway, and coal-heavy regions). Validate your function against the Patterson et al. [2021] estimates for GPT-3 training. Then use it to estimate the carbon cost of training a hypothetical model on 1,024 H100 GPUs for 90 days in each of the four regions, and produce a bar chart comparing the results.

References

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? *Proceedings of FAccT*, 610–623.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016). Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. *Advances in NeurIPS*.

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics Derived Automatically from Language Corpora Contain Human-like Biases. *Science*, 356(6334), 183–186.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., Oprea, A., and Steinke, T. (2021). Extracting Training Data from Large Language Models. *Proceedings of USENIX Security Symposium*.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daume III, H., and Crawford, K. (2021). Datasheets for Datasets. *Communications of the ACM*, 64(12), 86–92.

Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. (2022). Deduplicating Training Data Makes Language Models Better. *Proceedings of ACL*, 8424–8445.

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., and Gebru, T. (2019). Model Cards for Model Reporting. *Proceedings of FAccT*, 220–229.

Nadeem, M., Bethke, A., and Reddy, S. (2021). StereoSet: Measuring Stereotypical Bias in Pretrained Language Models. *Proceedings of ACL*, 5356–5371.

Parrish, A., Chen, A., Nangia, N., Padmakumar, V., Phang, J., Thompson, J., Htut, P. M., and Bowman, S. R. (2022). BBQ: A Hand-Built Bias Benchmark for Question Answering. *Findings of ACL*, 2086–2105.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., and Dean, J. (2021). Carbon Emissions and Large Neural Network Training. arXiv:2104.10350.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3), 379–423.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP. *Proceedings of ACL*, 3645–3650.