

# Multi-Modal Performance Prediction for Next-Generation AI Model Serving Platforms

Conal Halley

Department of Computer Science, University of New Hampshire, Durham, NH, USA.  
conalh@unh.edu

Kazimierz Ciastek

Department of Computer Science, Binghamton University, Binghamton, NY, USA.  
kazimierzmail@binghamton.edu

Tonya Marsh

Department of Computer Science, George Mason University, Fairfax, VA, USA.  
tonyamarsh@gmu.edu

## Abstract

Next-generation artificial intelligence model serving platforms must manage heterogeneous accelerators, dynamic request mixtures, increasingly large foundation models, and stringent service-level expectations across cloud, edge, and hybrid infrastructures. Performance prediction in such environments can no longer rely on a single latency or throughput metric. It must become multi-modal, combining telemetry from hardware counters, request traces, model metadata, scheduler state, network conditions, energy usage, and governance signals. This paper examines multi-modal performance prediction as a socio-technical capability for AI serving platforms. It argues that prediction quality depends less on isolated algorithms than on architectural observability, data contracts, deployment feedback loops, and policy-aware governance. The discussion reviews foundational constraints in warehouse-scale and inference serving systems, then analyzes how architectural layering, feature modalities, modeling choices, fairness, robustness, and sustainability interact. It considers trade-offs among accuracy, interpretability, responsiveness, and operational cost. The paper also explores cross-domain comparisons with stream processing, edge computing, and federated learning, where similar tensions appear between centralized prediction and local autonomy. A central claim is that multi-modal prediction should be treated as a platform service rather than a model artifact, with explicit interfaces, versioning, auditability, and fallback behavior. Such treatment supports capacity planning, autoscaling, carbon-aware scheduling, incident response, and equitable service delivery. The conclusion outlines research directions for open benchmarks, uncertainty-aware prediction, and governance frameworks that align technical performance with broader institutional accountability.

## Keywords

multi-modal prediction, AI model serving, inference platforms, performance modeling, observability, sustainability, fairness, governance, edge computing, systems architecture.

## 1. Introduction

Artificial intelligence serving platforms have become critical infrastructure for organizations that depend on real-time prediction, generative capabilities, recommendation, and decision support. Unlike traditional application servers, these platforms host models whose

computational profiles vary dramatically with input shape, sequence length, batch composition, accelerator type, memory pressure, and concurrent workload. A request that appears simple at the application layer may trigger substantial matrix operations, memory movement, kernel launches, and cross-device communication. Consequently, performance prediction has emerged as a central operational problem for capacity planning, autoscaling, admission control, and service-level management. Early warehouse-scale computing research established that datacenter performance is shaped by the interaction of hardware, software, and workload variability rather than by any single component [1]. Later work on tail latency demonstrated that rare slow requests can dominate user experience and require coordinated mitigation across replicated services [2]. These insights remain directly relevant to AI serving, where inference latency distributions are often heavy-tailed and highly sensitive to batching decisions, cache state, and scheduling policy.

Modern serving systems attempt to automate deployment and resource allocation through model-less abstractions and latency-aware scheduling. Platforms such as INFaaS and Clipper showed that serving infrastructure can decouple model selection from model execution, enabling per-query optimization and resource sharing [3][4]. More recent work has explored predictable execution and memory-efficient serving for large language models, where throughput and latency depend on token-level batching, key-value cache management, and memory fragmentation [5][6]. Yet many operational decisions still rely on coarse heuristics, historical averages, or single-metric thresholds. A recent conference contribution demonstrated that machine learning algorithms can predict large language model inference delay from observable features with useful accuracy [7]. That finding is important, but it also raises broader questions. If prediction models are embedded in serving platforms, how should they be trained, validated, governed, and updated? What data modalities are necessary? How should prediction errors be handled when they affect user-facing service levels or energy consumption?

This paper develops a system-level account of multi-modal performance prediction for next-generation AI model serving platforms. The term multi-modal here refers to the integration of multiple data sources and signal types, not merely to text, image, and audio modalities. These sources include infrastructure telemetry, request context, model configuration, runtime scheduler state, network conditions, energy measurements, and policy constraints. The paper argues that performance prediction should be understood as a platform capability that sits at the intersection of systems engineering, machine learning, and governance. It must support not only accurate estimates but also interpretability, robustness, fairness, and sustainability. The following sections examine foundational constraints, architectural layers, data modalities, modeling trade-offs, governance implications, deployment concerns, and future research directions.

## **2. Foundations of Multi-Modal Performance Prediction**

Performance prediction for AI serving begins with the recognition that inference is not a monolithic operation. It is a pipeline of preprocessing, tokenization or feature transformation, model execution, postprocessing, and network transfer. Each stage can become a bottleneck depending on workload and infrastructure. Research on transformer inference scaling has shown that latency and throughput are affected by model parallelism, batch size, sequence length, and memory bandwidth [8]. Large-scale training and serving studies further demonstrate that communication overhead, accelerator topology, and scheduling choices can

dominate performance at scale [9]. These findings imply that prediction models must incorporate architectural context rather than treating models as black boxes with fixed costs.

A second foundation is the inherent variability of production workloads. Requests arrive in bursts, vary in priority, and may be distributed across regions or tenants. Multi-tenant platforms must balance isolation, fairness, and utilization. A prediction model trained on average load may perform poorly during incidents, flash crowds, or partial failures. Therefore, multi-modal prediction must estimate not only expected performance but also uncertainty and sensitivity to changing conditions. The tail-at-scale perspective suggests that rare events matter disproportionately, especially when service-level objectives are defined at high percentiles [2]. Prediction systems should therefore represent distributional information, not merely point estimates.

A third foundation is the socio-technical nature of performance. AI serving platforms are embedded in organizations with competing goals. Engineering teams seek reliability and low latency, finance teams seek cost efficiency, sustainability teams seek carbon reduction, and compliance teams seek auditability. Energy and policy research has shown that large neural network development and deployment have significant environmental implications [10][11]. Prediction models that ignore these dimensions may optimize one objective while degrading others. For example, aggressive batching can improve accelerator utilization but increase latency for interactive users. Similarly, routing requests to distant regions may reduce cost but increase network energy and latency. Multi-modal prediction can help expose these trade-offs if it incorporates energy, carbon intensity, and policy signals alongside conventional performance metrics.

Finally, performance prediction must contend with technical debt. Machine learning systems accumulate hidden dependencies, feedback loops, and undeclared data contracts [12]. A prediction service that consumes telemetry from many subsystems can easily become fragile if those signals change. Therefore, the design of prediction infrastructure must include versioning, schema validation, drift detection, and reproducible evaluation. Without such discipline, prediction models become another source of operational uncertainty rather than a tool for reducing it.

### **3. Architectural Layers and Platform Constraints**

A next-generation AI serving platform can be understood through several interacting layers. The infrastructure layer includes compute, memory, storage, and network resources. The runtime layer includes model executors, compilers, memory managers, and schedulers. The service layer includes request routing, admission control, caching, and autoscaling. The governance layer includes access control, auditing, fairness constraints, and compliance reporting. Multi-modal performance prediction must operate across these layers. A prediction model that observes only runtime metrics may miss storage bottlenecks or policy-induced throttling. Conversely, a model that observes only business-level request attributes may fail to capture hardware-level contention.

Memory management illustrates the depth of the problem. Large language model serving requires efficient key-value cache handling, and systems such as PagedAttention have shown that memory fragmentation and sharing can significantly affect throughput [6]. Prediction models must therefore incorporate cache occupancy, sequence length distribution, and eviction behavior. Similarly, predictable execution frameworks have demonstrated that careful scheduling and resource isolation can reduce variance in deep neural network

inference [5]. This suggests that prediction and control are tightly coupled. A prediction service can inform scheduling, but scheduling decisions also change the performance signals that the prediction service observes. This feedback loop can create instability if not managed carefully.

Autoscaling and provisioning present another architectural constraint. Inference pipelines often contain multiple stages with different resource profiles. InferLine and related work on latency-aware provisioning show that end-to-end latency depends on pipeline configuration, replica counts, and batching policies [15]. A multi-modal predictor must estimate performance for candidate configurations, not just current ones. This requires counterfactual reasoning and simulation-like capabilities. In practice, platforms may combine lightweight analytical models with machine learning predictors. Analytical models provide rough bounds, while learned models capture complex interactions. The challenge is to integrate them without creating opaque decision paths that operators cannot debug.

Stream processing research offers a useful comparison. Systems that automatically scale distributed streaming dataflows must make rapid decisions under uncertain load [16]. They often rely on queueing approximations, operator profiles, and feedback control. AI serving platforms face similar dynamics, but with greater model heterogeneity and stricter latency requirements. The architectural lesson is that prediction should be embedded in a control loop with explicit objectives, constraints, and fallback policies. It should not be a standalone dashboard. When predictions are uncertain, the platform should default to conservative resource allocation or graceful degradation. When predictions are confident, the platform can pursue efficiency gains through consolidation, batching, or carbon-aware routing.

#### **4. Data Modalities and Feature Engineering for Prediction**

Multi-modal prediction depends on the availability and quality of data modalities. The first modality is infrastructure telemetry. This includes accelerator utilization, memory bandwidth, cache hit rates, power draw, temperature, and network throughput. The second modality is request context. This includes input size, sequence length, modality type, tenant identity, priority class, and historical behavior. The third modality is model metadata. This includes architecture family, parameter count, quantization level, compilation target, and memory footprint. The fourth modality is runtime state. This includes batch size, queue depth, active sequences, cache occupancy, and scheduler decisions. The fifth modality is environmental and policy data. This includes region, carbon intensity, data residency constraints, and fairness policies.

Each modality has distinct temporal properties. Hardware counters change rapidly, while model metadata changes only when deployments occur. Request context arrives with each request, while policy constraints may change slowly or episodically. A prediction system must align these streams without losing temporal relationships. Simple concatenation of features can be misleading if signals have different sampling rates or latencies. For example, a power reading may lag behind a burst of requests, causing the predictor to attribute latency to an event that has already passed. Therefore, feature engineering must include time alignment, windowing, and missing-data handling. It must also account for observability overhead. Collecting fine-grained telemetry can itself consume resources and perturb performance.

Privacy and security add further constraints. Tenant request data may be sensitive, and infrastructure telemetry may reveal business patterns. A prediction service should therefore operate under data minimization, access control, and aggregation policies. Federated or split

learning approaches may be relevant when prediction models must be trained across organizational boundaries [16][17]. However, federation introduces communication costs and heterogeneity challenges. In some cases, synthetic workloads or simulation traces may be preferable to raw request data. The choice depends on the sensitivity of the data, the required prediction accuracy, and the governance context.

The recent work on delay prediction using machine learning algorithms illustrates both the promise and the limits of feature-driven approaches [7]. Such models can learn relationships between observable runtime features and inference delay. However, their performance depends on the representativeness of training data, the stability of the serving stack, and the granularity of telemetry. If the platform changes its batching policy, quantization method, or hardware allocation, the predictor may drift. Multi-modal prediction must therefore include continual evaluation and adaptation. It should not assume that a model trained on one deployment remains valid indefinitely.

## **5. Modeling Approaches and Trade-Offs**

The modeling layer of multi-modal performance prediction includes statistical models, machine learning models, simulation, and hybrid approaches. Statistical models such as queueing approximations are interpretable and computationally cheap. They can provide fast estimates for capacity planning and admission control. However, they often rely on assumptions that are violated in AI serving, such as stationary arrival processes or homogeneous service times. Machine learning models can capture nonlinear interactions among features, but they require training data, maintenance, and validation. Deep learning models may improve accuracy when large amounts of telemetry are available, but they can be difficult to explain and expensive to run. A practical platform may use a hierarchy of predictors: fast approximate models for online decisions and more expensive models for offline planning.

A central trade-off is between accuracy and responsiveness. Online autoscaling requires predictions within milliseconds, while offline capacity planning can tolerate longer computation. Another trade-off is between centralization and locality. A centralized predictor can leverage cross-tenant data and global visibility, but it may introduce latency, privacy risks, and single points of failure. Local predictors can adapt to edge conditions and reduce data movement, but they may suffer from limited data and inconsistent updates. Edge computing research has explored similar tensions, showing that computation placement affects latency, bandwidth, and energy [14]. AI serving platforms should therefore support hierarchical prediction, where local models handle fast decisions and central models provide periodic calibration.

Interpretability is another important dimension. Operators need to understand why a prediction was made, especially when it affects service levels or cost. Model cards, datasheets, and documentation practices offer useful templates for transparency [13][14]. A prediction service should expose feature importance, uncertainty ranges, and known limitations. It should also record prediction provenance, including the data sources and model version used. This is essential for incident review and regulatory compliance. An opaque predictor that cannot be audited may be unacceptable in high-stakes domains such as healthcare, finance, or public services.

Compression and efficiency techniques also interact with prediction. Quantization, pruning, and distillation can reduce model size and latency, but they change performance

characteristics [17]. A predictor trained on full-precision models may not generalize to quantized variants. Similarly, compiler optimizations and kernel fusion can shift bottlenecks from compute to memory or communication [8]. Multi-modal prediction must therefore be aware of the model lifecycle. When models are updated, compiled, or quantized, the prediction system should be retrained or recalibrated. This suggests a need for continuous integration and continuous delivery pipelines that include performance prediction validation as a first-class stage.

## **6. Governance, Fairness, and Policy Implications**

Performance prediction is not politically neutral. It allocates resources, shapes user experience, and influences energy consumption. If prediction models systematically underestimate the needs of certain tenants, regions, or use cases, they can create unfair service degradation. For example, a platform that optimizes for average latency may deprioritize low-volume but socially important workloads. A platform that routes requests based on carbon intensity may inadvertently disadvantage regions with limited renewable energy. Governance frameworks must therefore define fairness objectives and evaluation criteria for prediction systems. These criteria should include error distribution across groups, not just aggregate accuracy.

Fairness in this context is multidimensional. It includes allocative fairness, meaning fair distribution of resources; procedural fairness, meaning transparent and contestable decision processes; and representational fairness, meaning that prediction models do not encode historical biases. Model cards and datasheets provide documentation mechanisms, but they must be complemented by ongoing monitoring [13][14]. Prediction models should be audited for disparate error rates and for feedback loops that reinforce disadvantage. If a tenant receives lower priority because of historical predictions, its future performance may worsen, creating a self-fulfilling prophecy. Governance should include appeal mechanisms and periodic review.

Policy implications extend to environmental sustainability. Carbon-aware computing seeks to shift workloads to times and places with cleaner energy. Multi-modal prediction can support this by forecasting latency, energy, and carbon intensity together. However, carbon-aware scheduling must be balanced against service-level objectives and data residency requirements. If a request is delayed too long for carbon reasons, users may lose trust. If it is routed across borders, legal constraints may be violated. Therefore, policy-aware prediction should treat carbon, latency, cost, and compliance as joint constraints. It should also provide clear explanations when trade-offs are made.

Security and robustness are part of governance as well. A prediction service that relies on telemetry can be attacked through data poisoning or signal spoofing. If an adversary manipulates queue depth or power readings, the predictor may make harmful scaling decisions. Platforms should therefore validate inputs, detect anomalies, and maintain fallback policies. The same applies to model outputs. A prediction that is uncertain or out-of-distribution should trigger conservative behavior rather than aggressive optimization. Governance frameworks should specify who is accountable when automated predictions lead to outages, unfair treatment, or excessive emissions.

## **7. Deployment, Robustness, and Sustainability**

Deploying multi-modal prediction in production requires careful attention to robustness. Prediction services should be redundant, observable, and degradable. If the predictor fails, the platform should revert to static policies or conservative defaults. If telemetry is missing, the

predictor should indicate increased uncertainty rather than silently imputing values. If the platform is under attack or experiencing a major incident, prediction may need to be disabled in favor of simple heuristics. These design choices should be documented and tested through chaos engineering and simulation.

Sustainability is both an operational and ethical concern. Large-scale AI serving consumes significant energy, and the carbon intensity of electricity varies by region and time [10][11]. Multi-modal prediction can help by estimating the energy cost of alternative scheduling decisions. For example, a platform might delay non-urgent batch jobs until renewable energy is abundant, while prioritizing interactive requests. It might also choose more efficient hardware configurations for certain model types. However, such optimization requires accurate energy models. Measuring energy at the accelerator, server, and datacenter levels is challenging, and virtualization can obscure attribution. Prediction systems should therefore incorporate uncertainty in energy estimates and avoid overclaiming precision.

Technical debt remains a major risk. A prediction service that depends on many data sources can become brittle if those sources change. Schema evolution, clock skew, missing metrics, and inconsistent labeling can degrade performance. Teams should treat prediction features as contracts with versioning and validation [12]. They should also monitor prediction drift and retrain models when necessary. In some cases, simpler models may be more sustainable than complex ones because they require less maintenance and computation. The goal is not maximum sophistication but fit-for-purpose reliability.

Cross-domain comparisons can illuminate deployment strategies. Federated learning systems must balance local adaptation with global coordination, often under heterogeneous data and resource constraints [16]. Edge AI systems must decide what to compute locally versus remotely, considering latency, bandwidth, privacy, and energy [14]. Stream processing systems must scale rapidly while maintaining exactly-once semantics and low latency [16]. AI serving platforms share these concerns but also face unique model-level variability. A multi-modal predictor that works well for one domain may need substantial adaptation for another. Therefore, reusable architectures should emphasize modular feature pipelines, common telemetry standards, and pluggable prediction models.

## **8. Case Illustrations and Cross-Domain Comparisons**

Consider a cloud platform that serves both small recommendation models and large generative models. The small models may be latency-sensitive and highly cacheable, while the large models may benefit from batching and memory pooling. A single prediction model for all workloads would likely fail. A multi-modal approach could classify requests by model family, input size, and priority, then route them to specialized predictors. The platform could use historical traces to estimate queueing delay, hardware counters to estimate compute time, and memory metrics to estimate cache behavior. When predictions are uncertain, the platform could reserve extra capacity or shed low-priority load.

In an edge deployment, the same platform might run across devices with limited power and intermittent connectivity. Multi-modal prediction would need to incorporate battery state, thermal conditions, network quality, and user context. A prediction that minimizes latency might drain the battery, while one that minimizes energy might miss interactive deadlines. The platform could use local predictors for fast decisions and periodically synchronize with a central model. Federated learning could help improve local models without sharing raw user

data [16]. However, the central model may not generalize to every device, so local validation is essential.

Infrastructure as a service and serverless platforms offer another comparison. Serverless inference abstracts away servers but introduces cold starts, concurrency limits, and billing granularity. Prediction models must estimate warm-up time, concurrency effects, and cost. Research on model serving systems such as Clipper and INFaaS shows that abstraction can improve flexibility but also creates new scheduling challenges [3][4]. A multi-modal predictor could combine function metadata, historical cold-start data, and current cluster load. It could then decide whether to reuse a warm container, provision a new one, or route to another region.

Benchmarking is essential for comparing these approaches. MLPerf and related efforts provide standardized workloads and metrics for inference performance [20]. However, benchmarks often focus on throughput and latency under controlled conditions. Multi-modal prediction requires additional dimensions, including energy, carbon, fairness, and robustness. Future benchmarks should include realistic multi-tenant workloads, adversarial conditions, and governance constraints. They should also report uncertainty and failure modes, not just average performance. Without such benchmarks, claims about prediction quality remain difficult to compare across platforms.

## **9. Future Directions**

Several research directions follow from this analysis. First, platforms need standards for telemetry and metadata that support multi-modal prediction without excessive overhead. These standards should define common schemas for request context, hardware counters, model metadata, and policy signals. They should also specify privacy-preserving aggregation methods. Second, prediction systems should incorporate uncertainty quantification and out-of-distribution detection. Operators need to know when a prediction is reliable and when it should be ignored. Third, research should explore hierarchical and federated prediction architectures that balance global learning with local adaptation. This is particularly important for edge and multi-cloud deployments.

Fourth, governance frameworks must evolve alongside technical capabilities. Auditing, certification, and accountability mechanisms should cover prediction models as well as the models they serve. Regulators and standards bodies may need to define expectations for fairness, transparency, and environmental reporting. Fifth, sustainability should be integrated into prediction objectives from the beginning, not added as an afterthought [10][11]. This requires better energy measurement, carbon-aware scheduling, and lifecycle assessment. Sixth, platforms should support simulation and digital twins for counterfactual analysis. Operators could test scheduling policies and prediction models against synthetic workloads before deploying them in production.

Finally, interdisciplinary collaboration is necessary. Systems researchers, machine learning scientists, social scientists, legal scholars, and policymakers must work together to design prediction systems that are technically effective and socially legitimate. The required reference on machine learning-based delay prediction [7] points to a promising technical direction, but it also highlights the need for broader evaluation. Accuracy alone is insufficient if the system is opaque, unfair, brittle, or unsustainable. The next generation of AI serving platforms will be judged by their ability to deliver reliable performance under diverse conditions while respecting environmental and societal constraints.

## 10. Conclusion

Multi-modal performance prediction is becoming a foundational capability for AI model serving platforms. As models grow larger, workloads become more heterogeneous, and infrastructure becomes more distributed, simple latency or throughput estimation is no longer adequate. Platforms must integrate telemetry, request context, model metadata, runtime state, energy data, and policy constraints into prediction systems that support autoscaling, scheduling, admission control, and sustainability. This paper argued that such prediction should be treated as a platform service rather than an isolated model. It requires explicit interfaces, versioning, observability, uncertainty management, and governance. The technical trade-offs among accuracy, interpretability, responsiveness, and cost must be managed through layered architectures and fallback policies. The socio-technical trade-offs among efficiency, fairness, robustness, and environmental impact must be managed through transparent governance and accountability.

The discussion also emphasized that performance prediction is not merely a modeling problem. It is an architectural and institutional problem. Data contracts, telemetry standards, model lifecycle management, and auditing practices determine whether prediction systems remain reliable over time. Cross-domain comparisons with stream processing, edge computing, and federated learning show that similar challenges appear in other distributed systems, but AI serving platforms face unique model-level variability and rapid innovation cycles. Future work should develop open benchmarks, uncertainty-aware predictors, hierarchical deployment patterns, and policy frameworks that align technical optimization with societal values. If designed well, multi-modal performance prediction can help platforms deliver efficient, equitable, and sustainable AI services at scale.

## References

1. Barroso, L. A., Clidaras, J., & Hölzle, U. (2013). *The datacenter as a computer: An introduction to the design of warehouse-scale machines* (2nd ed.). Morgan & Claypool.
2. Dean, J., & Barroso, L. A. (2013). The tail at scale. *Communications of the ACM*, 56(2), 74-80. <https://doi.org/10.1145/2408776.2408794>
3. Romero, F., Li, Q., Govindaraju, V., & Guo, Y. (2021). INFaaS: Automated model-less inference serving. In *2021 USENIX Annual Technical Conference* (pp. 397-411). USENIX Association.
4. Crankshaw, D., Wang, X., Zhou, G., Franklin, M. J., Gonzalez, J. E., & Stoica, I. (2017). Clipper: A low-latency online prediction serving system. In *14th USENIX Symposium on Networked Systems Design and Implementation* (pp. 613-627). USENIX Association.
5. Gujarati, A., Karimi, R., Alzayat, S., Hao, W., Kaufmann, A., Vigfusson, Y., & Mace, J. (2020). Serving DNNs like clockwork: Performance predictability from the bottom up. In *14th USENIX Symposium on Operating Systems Design and Implementation* (pp. 443-462). USENIX Association.
6. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., & Stoica, I. (2023). Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles* (pp. 611-626). ACM.

7. Zhang, J., & Gao, H. (2026, March). LLM inference delay prediction based on machine learning algorithms. In 2026 International Conference on Generative Artificial Intelligence and Information Security (GAIIS) (pp. 37-41). IEEE.
8. Pope, R., Douglas, S., Chowdhery, A., Devlin, J., Bradbury, J., Heek, J., Xiao, K., Agrawal, S., & Dean, J. (2023). Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems*, 5, 606-624.
9. Narayanan, D., Shoeybi, M., Casper, J., LeGresley, P., Patwary, M., Korthikanti, V., Vainbrand, D., Kashinkunti, P., Bernauer, J., Catanzaro, B., Phanishayee, A., & Zaharia, M. (2021). Efficient large-scale language model training on GPU clusters using Megatron-LM. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 1-15). ACM.
10. Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., So, D., Texier, M., & Dean, J. (2021). Carbon emissions and large neural network research. *arXiv preprint arXiv:2104.10350*.
11. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3645-3650). ACL.
12. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J. F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems 28* (pp. 2503-2511). Curran Associates.
13. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (pp. 220-229). ACM.
14. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., & Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM*, 64(12), 86-92.
15. Crankshaw, D., Sela, O., Mo, X., Zumar, C., Stoica, I., Gonzalez, J. E., & Tumanov, A. (2020). InferLine: Latency-aware provisioning and scheduling for machine learning inference pipelines. In *Proceedings of the 11th ACM Symposium on Cloud Computing* (pp. 1-14). ACM.
16. Kalavri, V., Liagouris, J., Hoffmann, M., Dimitrova, D., Forshaw, M., & Roscoe, T. (2018). Three steps is all you need: Fast, accurate, automatic scaling decisions for distributed streaming dataflows. In *13th USENIX Symposium on Operating Systems Design and Implementation* (pp. 783-798). USENIX Association.
17. Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems 35* (pp. 30318-30332). Curran Associates.
18. Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., ... & Kaplan, J. (2022). Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

19. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (pp. 610-623). ACM.
20. Reddi, V. J., Cheng, C., Kanter, D., Mattson, P., Schmuelling, G., Wu, C. J., Anderson, B., Gu, W., Bhandare, A., Sokolov, A., & Tang, H. (2020). MLPerf inference benchmark. In 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (pp. 446-459). IEEE.