**ORIGINAL RESEARCH** **Open access**

# The Scaffolding Problem: How Materials AI Depends on Unstated Scientific Infrastructure

Paolo Ricci[1*], Marco De Luca[1], Giulia Ferraro[2], Antonio Russo[1]

## Abstract

The scaffolding problem represents a critical yet underrecognized failure mode in artificial intelligence for materials science, wherein seemingly autonomous AI systems depend on vast, unstated layers of scientific infrastructure that remain invisible until they fail. At this point, the AI models themselves cease to function reliably or reproducibly. This failure mode arises because materials AI pipelines are not standalone artifacts but are instead supported by extensive scaffolding—databases, software ecosystems, computational resources, measurement standards, tacit community knowledge, and institutional frameworks—that enable data ingestion, model training, and inference but are rarely documented or maintained as core components of the research. The scaffolding problem is formally defined as the failure to recognize, document, and sustain these invisible support structures, leading to unrecognized vulnerabilities that undermine the reliability of data-driven discoveries in materials design. Materials AI depends on six distinct types of scientific infrastructure, ranging from data repositories such as the Materials Project and AFLOW to software libraries like pymatgen and institutional funding mechanisms, each of which carries hidden assumptions about stability and accessibility. Scaffolding failures occur through mechanisms including infrastructure decay, dependency drift, access loss, and knowledge erosion, producing a typology of four specific failure modes: silent dependency failure, reproducibility collapse, infrastructure lock-in, and knowledge gap failure. Detection relies on systematic dependency mapping, version pinning, access monitoring, reproduction testing, and knowledge auditing, while mitigation demands explicit documentation, containerization, data archiving, dependency minimization, infrastructure independence, and knowledge capture. By articulating the scaffolding problem as a distinct failure mode, this analysis reveals how unexamined infrastructure dependencies threaten the long-term viability of materials AI and calls for a fundamental shift toward treating scaffolding as an explicit, first-class concern in research practice.

**Keywords** Materials AI, Failure mode analysis, Scaffolding problem, Scientific infrastructure, Infrastructure decay, Dependency drift

*Correspondence:
Paolo Ricci
paolo.ricci@gmail.com
[1] Department of Materials Science and AI Systems, University of Naples Federico II, Naples, Italy
[2] Department of Smart Materials Analytics, University of Bologna, Bologna, Italy

## Introduction

Materials AI systems do not stand alone. They depend on vast infrastructure—databases, software libraries, computational clusters, measurement standards, tacit knowledge—that functions as invisible scaffolding supporting every stage from data curation to model deployment. This scaffolding is rarely acknowledged in publications or code repositories, creating the illusion of self-contained, robust AI pipelines when in reality the systems are deeply entangled with external resources whose stability is assumed rather than guaranteed. When this scaffolding fails, the AI systems fail in ways that are difficult to diagnose because the root causes lie outside the model architecture or training algorithm itself. This paper analyzes the scaffolding problem as a distinct failure mode in materials AI, drawing on foundational concepts from the ecology of infrastructure to illuminate how unstated dependencies shape the reliability and reproducibility of data-driven materials discovery [1-3].

The rapid adoption of machine learning in materials science has produced impressive advances in property prediction, materials screening, and inverse design [4, 5]. Yet these successes rest upon an extensive but largely invisible foundation. Databases such as the Materials Project provide curated datasets of computed properties, while frameworks like AFLOW enable high-throughput calculations that feed into AI training pipelines [6, 7]. Software toolkits, including Matminer, facilitate feature engineering and data mining, and institutional resources supply the computational power necessary for training large models [8]. These elements are not peripheral; they constitute the very conditions of possibility for materials AI. Nevertheless, research narratives typically foreground the novelty of the AI algorithm while treating the supporting infrastructure as given and unproblematic.

This omission is not merely a matter of rhetorical convenience. It reflects a deeper epistemological stance in data-driven science, whereby infrastructure is rendered transparent precisely because it works [1]. As long as databases remain accessible, software dependencies resolve cleanly, and computational resources are available, the scaffolding remains unseen. The moment any component degrades—through bit rot in archived data, breaking changes in library APIs, or loss of access to proprietary measurement protocols—the entire pipeline can collapse in subtle or catastrophic ways. Such failures are especially pernicious in materials AI because the domain combines high-stakes applications (energy storage, catalysis, semiconductors) with extreme sensitivity to data quality and provenance. A model trained on data from a now-degraded database may produce predictions that appear numerically stable. Yet, it rests on corrupted foundations, leading to downstream experimental failures that are difficult to trace [3].

The scaffolding problem, therefore, demands analysis as a failure mode in its own right, distinct from algorithmic shortcomings or data-quality issues. It concerns the systemic invisibility of the support structures that make AI possible. By failing to document and maintain these structures, the field accumulates unrecognized technical and epistemic vulnerabilities [2]. The present work addresses this gap by providing a conceptual framework for understanding the scaffolding problem, its manifestations in materials AI, and pathways toward more robust practice. It begins by formally defining the problem, delineates the types of infrastructure involved, examines the mechanisms through which failures occur, and presents a typology of failure modes. Subsequent sections address detection and mitigation strategies, relations to other failure modes, and implications for research practice. Throughout, the analysis emphasizes that scaffolding is not a background condition but an integral, active component of materials AI systems whose neglect constitutes a genuine failure mode [3].

As illustrated in **Figure 1**, the hierarchical structure of materials AI systems shows how foundational layers—ranging from institutional and knowledge infrastructure up through measurement, computational, software, and data infrastructure—collectively support the AI model pipeline and ultimately determine system outputs. At the same time, instabilities at lower levels propagate upward, leading to issues such as irreproducibility and model unreliability.

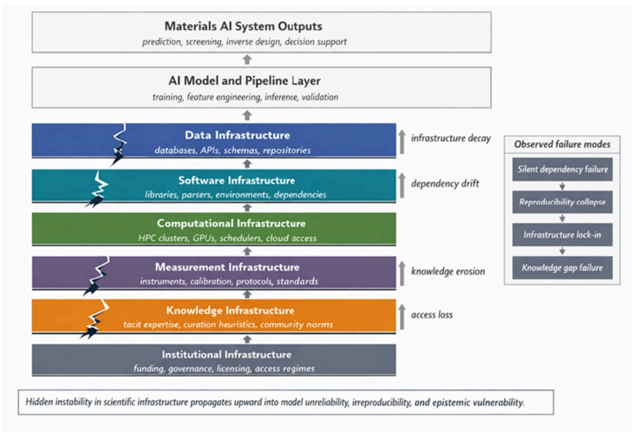


**Figure 1.** Hierarchical architecture of the scaffolding problem in materials AI.

Conceptually, the scaffolding problem can be visualized as a multi-layered architectural diagram in which the AI model occupies the topmost layer, supported by successive strata of data infrastructure, software infrastructure, computational infrastructure, measurement infrastructure, knowledge infrastructure, and institutional infrastructure. Breaks or erosions appear as jagged fractures propagating upward from lower layers, illustrating how an undetected failure at the database level can silently undermine model predictions without triggering obvious error messages. This layered representation highlights both the interdependence of the strata and the cumulative nature of risk when any single layer is left unexamined.

The urgency of addressing the scaffolding problem stems from the accelerating scale of materials AI research. With the proliferation of large-scale databases and foundation models, dependency chains grow longer and more opaque [9-12]. At the same time, the push toward open science and reproducibility creates pressure to ensure that published results remain verifiable years or decades later. Infrastructure that appears permanent today may prove ephemeral tomorrow due to funding shifts, institutional changes, or technological obsolescence. Recognizing the scaffolding problem, therefore, requires a shift in perspective: from viewing AI systems as isolated intellectual achievements to understanding them as socio-technical assemblages whose reliability hinges on the sustained health of their supporting infrastructure [1]. This paper offers the conceptual tools necessary for that shift.

# Defining the Scaffolding Problem

The scaffolding problem arises at the intersection of data-driven science and the sociology of scientific infrastructure. It captures the systematic failure to recognize, document, and maintain the extensive but unstated scientific infrastructure upon which materials AI systems depend. Formally, the scaffolding problem is the failure to recognize, document, and maintain the scientific infrastructure on which AI systems depend, leading to unrecognized vulnerabilities and reproducibility failures in materials AI pipelines.

This definition draws directly from the ecological perspective on infrastructure articulated by Star and Ruhleder, who emphasize that infrastructure becomes visible only when it breaks [1]. In materials AI, the scaffolding remains invisible during routine operation precisely because it functions reliably; the moment it degrades, the failure is misattributed to the model itself rather than to the underlying support structure [3].

It is essential to distinguish the scaffolding problem from related concepts such as technical debt. Technical debt, as originally formulated in the machine-learning context, refers to accumulated compromises in code quality, data pipelines, and model maintenance that increase long-term costs [2]. Scaffolding failure, by contrast, operates at a broader socio-technical scale. It encompasses not only code-level issues but also external databases, institutional resources, and tacit knowledge that lie outside the immediate control of any single research team. Whereas technical debt is typically internal to a project's codebase, scaffolding concerns the external dependencies whose maintenance is distributed across communities, funding agencies, and infrastructure providers [3].

The scaffolding problem is also distinct from simple data-quality concerns. Poor data quality can be detected and mitigated within the modeling workflow itself. Scaffolding failure, however, concerns the conditions that make data available and usable in the first place. When the Materials Project database undergoes schema changes or when AFLOW's high-throughput computation pipeline is updated without backward compatibility, the impact propagates silently to every downstream AI model trained on those resources [6, 7]. These changes are not errors in the conventional sense; they are transformations of the infrastructure itself, yet they are rarely flagged as such in the materials AI literature [13-23].

Furthermore, the scaffolding problem possesses an epistemological dimension. It reflects unexamined assumptions about the permanence and universality of scientific resources. Researchers assume that the ICSD will remain accessible, that pymatgen will continue to parse legacy file formats, and that GPU clusters will be available under consistent queuing policies [8]. These assumptions become embedded in the design of AI pipelines, rendering the systems brittle when the assumptions prove false. The problem is therefore not merely technical but epistemic: it concerns the knowledge infrastructure that underpins what counts as valid materials AI research [1].

By formalizing the scaffolding problem in this manner, the present analysis establishes it as a failure mode that is both pervasive and diagnosable. It is pervasive because every material AI project, regardless of scale, rests on multiple layers of scaffolding. It is diagnosable because the failure leaves distinctive signatures once the invisibility is pierced —signatures that include sudden drops in model performance upon environment changes, inability to reproduce published results, and dependence on resources whose future availability is uncertain [3, 10]. Recognizing the scaffolding problem thus provides a new lens through which to evaluate the robustness of materials AI systems and to design interventions that treat infrastructure as an explicit object of concern rather than an implicit background condition.

# Types of Scientific Infrastructure

Materials AI depends on six distinct but interlocking types of scientific infrastructure, each of which operates largely outside the explicit purview of model developers yet exerts profound influence on system behavior. These types are rarely enumerated together in a single analysis, yet their collective presence constitutes the scaffolding that enables contemporary materials AI.

Data Infrastructure comprises the databases, data formats, and APIs that supply raw and curated materials data. Prominent examples include the Materials Project, AFLOW, OQMD, JARVIS, and NOMAD repositories, which host millions of computed and experimental entries [6, 7, 10, 17]. Data infrastructure is often unstated because researchers treat database queries as neutral data ingestion steps rather than as interactions with evolving, maintained systems whose schemas, versioning policies, and access controls can change without notice. Models trained on legacy data suddenly become irreproducible when a database deprecates an older API endpoint.

Software Infrastructure includes the libraries and frameworks—pymatgen, ASE, scikit-learn, PyTorch, TensorFlow, Matminer—that handle data processing, feature engineering, model training, and deployment [8, 23]. Version dependencies within these libraries create intricate compatibility graphs that are seldom fully documented. A seemingly minor update to a dependency can break feature extraction pipelines or alter numerical precision in ways that subtly degrade model outputs. Because software infrastructure is treated as a tool rather than as infrastructure, its maintenance costs and evolutionary dynamics remain externalized from the research narrative.

Computational Infrastructure encompasses HPC clusters, cloud computing resources, GPU availability, and associated queuing and scheduling systems. Materials AI training jobs frequently require thousands of GPU-hours, yet the underlying hardware provisioning, software stack curation, and energy constraints are rarely discussed [9]. When institutional access policies change or cloud credits expire, entire training workflows become stalled. Computational infrastructure is unstated because it is assumed to be a generic resource rather than a specialized, time-sensitive component whose configuration directly shapes which models can be trained and at what scale.

Measurement infrastructure consists of experimental standards, calibration protocols, instrument specifications, and data-acquisition pipelines that link computational predictions to physical reality. Although materials AI is often computation-centric, validation against experimental data relies on a consistent measurement infrastructure [4]. These protocols—standardized test conditions, calibration curves, instrument firmware—are seldom cited explicitly. When an instrument is recalibrated or a standard is updated, the ground-truth labels used for model training may shift, introducing discrepancies that are difficult to trace without explicit documentation of the measurement scaffolding.

Knowledge Infrastructure includes the tacit knowledge, community norms, unpublished methods, and undocumented assumptions that guide data curation, hyperparameter selection, and interpretation of results [1]. Experienced researchers know which database entries are reliable, which software quirks to avoid, and which experimental caveats apply, yet this knowledge is transmitted informally. As personnel move between institutions or retire, the knowledge infrastructure erodes. Materials AI inherits these tacit layers because the field has grown rapidly, outpacing formal codification of best practices [13, 24-26].

Institutional Infrastructure covers funding streams, journal access policies, institutional subscriptions to databases, collaborative networks, and governance structures that sustain the preceding five types [3]. Open-access mandates, grant requirements for data sharing, and university licensing agreements determine which databases remain available and under what conditions. These institutional factors are rarely mentioned in technical papers, yet they determine the very existence and longevity of the other infrastructure layers. When funding for a major repository ends or subscription costs rise, the downstream effects on materials AI are immediate yet rarely attributed to institutional scaffolding failure.

Each type of infrastructure carries implicit assumptions about continuity, universality, and neutrality. By rendering these assumptions explicit, the present analysis reveals how materials AI is structurally dependent on a heterogeneous socio-technical assemblage whose maintenance lies beyond any single research group.

**Table 1** consolidates the six infrastructure types into an analytical matrix that clarifies how each layer carries a distinct hidden assumption, vulnerability pathway, and empirical failure signature.

**Table 1.** Analytical matrix of scientific scaffolding in materials AI: infrastructure type, hidden assumption, vulnerability mechanism, and failure signature

| Infrastructure type | Core function in materials AI | A hidden assumption is typically left unstated | Domir vulnera mechan |
|---|---|---|---|
| Data infrastructure | Supplies curated materials records, metadata, labels, and queryable repositories | Databases are stable, queryable, version-consistent, and historically recoverable | Infrastru decay; a loss depend drif |
| Software infrastructure | Enables parsing, feature engineering, modeling, and reproducible execution | Libraries will remain backward compatible and numerically consistent across versions | Depend drif infrastru deca |
| Computational infrastructure | Provides compute, storage, schedulers, and hardware capacity for training and inference | Compute access is continuous, affordable, and technically interchangeable | Access depend drif |
| Measurement infrastructure | Stabilizes links between computational outputs and experimental ground truth | Calibration standards, protocols, and instruments remain comparable over time | Infrastru deca knowle erosi |
| Knowledge infrastructure | Carries tacit expertise, curation norms, and interpretation heuristics | Critical know-how is socially durable and transferable without formal capture | Knowle erosi |
| Institutional infrastructure | Sustains repositories, licenses, subscriptions, governance, and funding continuity | Supporting organizations will continue to maintain access and stewardship | Access knowle erosi |

# Mechanisms of Scaffolding Failure

Scaffolding failures in materials AI systems rarely arise as isolated anomalies; rather, they emerge through structured processes that unfold within the layered technical and epistemic infrastructure supporting model development and deployment. What appears at the surface as sporadic malfunction is more accurately understood as the cumulative effect of underlying mechanisms that degrade, misalign, restrict, and ultimately destabilize the conditions under which models operate. This perspective reframes failure not as an endpoint but as a trajectory shaped by the evolving relationships between data, tools, and human practices.

A central dynamic in this trajectory is the gradual deterioration of infrastructural integrity. Digital and physical resources do not remain static, and over time, they undergo subtle forms of degradation that accumulate without immediate visibility. Archived datasets experience bit rot, file formats lose compatibility, and indexing systems fall out of synchronization with contemporary standards [3]. Within materials science, such processes become evident when legacy entries in widely used repositories diverge from updated computational protocols, or when crystallographic files can no longer be parsed reliably by current software environments [6, 8]. Because this degradation unfolds incrementally, its effects are rarely attributed to a single identifiable moment. Instead, they surface indirectly, often during model retraining or benchmarking, where unexplained biases or inconsistencies begin to appear, resisting straightforward diagnosis.

Closely intertwined with this slow decay is a more dynamic process in which the evolution of upstream components destabilizes downstream assumptions. Computational ecosystems are inherently interdependent, and updates to libraries, data schemas, or numerical backends propagate through the system in ways that are not always anticipated [2]. In practice, even minor modifications—such as changes in tensor operations within widely used machine learning frameworks—can alter numerical behavior sufficiently to affect the convergence properties of models trained on established materials datasets [7, 23]. What makes this process particularly difficult to detect is its subtlety: models continue to train, outputs remain numerically plausible, and yet the alignment with physically validated benchmarks begins to erode. The system appears operational while its interpretive grounding quietly shifts.

Beyond these internal transformations, the stability of materials AI workflows is further contingent on continued access to the resources upon which they depend. When such access is disrupted, the consequences are often immediate but conceptually mischaracterized. Restrictions introduced through paywalls, the deprecation of APIs, the shutdown of databases, or the expiration of institutional licenses can abruptly sever the link between models and their foundational data sources [1]. A research group that previously relied on a specific subset of crystallographic data, for instance, may find itself unable to reproduce earlier results once that access is withdrawn. Although such events are frequently framed as issues of data availability, this framing underestimates their structural significance; what is lost is not merely data, but the continuity of the infrastructural environment within which scientific claims were generated [10].

Running parallel to these technical processes is a less visible but equally consequential form of degradation that operates at the level of community knowledge. Materials AI workflows depend not only on formalized procedures but also on tacit expertise—practices of data curation, parameter tuning, and interpretive judgment that are rarely fully documented. As researchers move between institutions, retire, or shift focus, this embedded knowledge dissipates, leaving gaps that are not easily reconstructed [13]. The effects become apparent when recurring challenges, once informally resolved within specific groups, are encountered anew by subsequent researchers. In this sense, knowledge erosion does not merely remove accumulated insight; it alters the conditions under which remaining practitioners engage with the infrastructure, often increasing their reliance on tools and datasets whose limitations are no longer fully understood. This reliance, in turn, amplifies susceptibility to the other forms of degradation already in motion [26].

The significance of these mechanisms lies not only in their individual effects but in the way they combine to produce compounded instability. A dataset that has undergone incremental degradation may simultaneously be subjected to changes in access conditions, while the tools required to interpret it evolve and the expertise needed to navigate these shifts diminishes. Under such conditions, failure is no longer attributable to a single source but emerges from the interaction of multiple processes that reinforce one another. Recognizing this interdependence shifts analytical attention away from isolated breakdowns toward the systemic patterns through which materials AI infrastructures become fragile, highlighting the need to understand failure as an emergent property of interconnected transformations rather than as a discrete event.

# A Typology of Scaffolding Failure Modes

The mechanisms outlined above do not remain abstract; they materialize in recurring configurations of failure that can be observed across diverse materials AI workflows. These configurations are best understood as archetypal modes through which infrastructural fragility becomes operationally visible, each characterized by a distinctive alignment between underlying mechanism and empirical signature. Rather than representing isolated breakdowns, they reflect patterned responses of the system to accumulated stress within its supporting scaffolding.

One such pattern emerges when degradation unfolds without interrupting system operation, allowing models to continue producing outputs that retain an appearance of plausibility while gradually diverging from physical reality. Under these conditions, failure does not announce itself through explicit errors but is embedded in the silent drift of predictions away from ground truth [3]. This dynamic is frequently associated with shifts in the underlying data or computational environment—whether through incremental database updates or subtle changes in dependencies—that remain unregistered within the model's training context. A graph neural network trained on an earlier snapshot of a materials database, for example, may persist in generating formation energy predictions that no longer correspond to revised reference values, leading to systematic misestimation of stability across specific chemical families. The absence of overt failure signals complicates detection, leaving discrepancies to surface only through careful comparison of training and validation distributions that cannot be reconciled through algorithmic explanations alone [6].

A more visible disruption occurs when the continuity of scientific claims is broken altogether, rendering previously reported results irreproducible despite nominally identical computational setups. In such cases, the failure is not gradual but abrupt, often triggered by the loss of access to critical resources or by shifts in software dependencies that invalidate prior configurations [2]. Materials AI provides numerous instances in which models that once reproduced benchmark properties—such as ionic conductivity in solid electrolytes—become inoperable following updates to foundational libraries or the disappearance of referenced datasets [8, 10]. The resulting condition is frequently attributed to the complexities of managing software versions, yet this explanation obscures the deeper infrastructural instability at play. What appears as a technical inconvenience is in fact a breakdown in the continuity of the scientific environment itself, where the linkage between code, data, and results can no longer be reconstructed.

Beyond these episodic failures lies a more entrenched form of vulnerability that develops through prolonged coupling between models and specific infrastructural components. Over time, systems may become so tightly integrated with particular data schemas, parsing routines, or computational environments that substitution becomes effectively infeasible. This condition does not arise from a single point of failure but from the absence of abstraction mechanisms that would otherwise allow components to be replaced without disrupting the entire system [23]. In materials AI contexts, such lock-in can manifest when models are tailored to proprietary data formats or interfaces, losing functionality as soon as those interfaces are modified or withdrawn. The presence of hard-coded assumptions within the codebase—assumptions that cannot be disentangled without retraining—serves as a clear diagnostic indicator of this state [12]. What is lost in these scenarios is not only flexibility but the capacity for adaptation, as the system's dependence on a specific scaffolding element constrains its future evolution.

A further layer of fragility becomes apparent when the implicit knowledge required to interpret and operate within the infrastructure is no longer available. Materials AI workflows often rely on tacit conventions—decisions about data filtering, normalization, or exclusion—that are embedded in practice rather than formalized in documentation. When this knowledge dissipates, either through personnel turnover or the absence of systematic capture, the assumptions underpinning model behavior become opaque to new users [1, 13]. The consequences can be abrupt and difficult to diagnose: models begin to generate outputs that violate basic physical constraints, yet appear internally consistent given the training data. A case in point arises when historically excluded classes of materials, such as certain high-pressure phases, are reintroduced without recognition of prior filtering conventions, leading to predictions that experienced practitioners would immediately question. The defining feature of this failure mode is the asymmetry of recognition, where errors are evident to those familiar with the lost context but remain invisible to those encountering the system anew.

These modes rarely occur in isolation. A system affected by silent drift may simultaneously become irreproducible as dependencies shift, while increasing reliance on specific infrastructure components deepens lock-in, and the erosion of tacit knowledge removes the interpretive safeguards that might otherwise mitigate emerging errors. Through this interplay, failure propagates across multiple dimensions, transforming localized weaknesses into systemic vulnerabilities. Framing these patterns as a typology does more than provide descriptive clarity; it establishes a conceptual vocabulary through which infrastructural breakdowns can be identified, compared, and ultimately anticipated. In doing so, it opens the possibility of treating such failures not as unavoidable byproducts of complexity but as structured phenomena amenable to intervention within the design and governance of materials AI systems.

**Table 2** differentiates the four scaffolding failure modes by showing that they vary not only in mechanism but also in visibility, diagnostic target, and intervention priority.

**Table 2.** Differentiating scaffolding failure modes in materials AI: diagnostic profile, level of visibility, and intervention priority

| Failure mode | Immediate manifestation | Typical visibility to researchers | Prima caus mechani |
|---|---|---|---|
| Silent dependency failure | Outputs remain plausible while underlying validity drifts | Low visibility; often mistaken for routine model variance | Infrastru deca depend drif |
| Reproducibility collapse | Published results cannot be regenerated despite nominally identical code | High visibility; usually appears only after replication attempts | Access depend drif |
| Infrastructure lock-in | The system cannot function once a specific database, toolchain, or environment changes | Medium visibility; recognized when substitution is attempted | Depend drift knowle erosi |
| Knowledge gap failure | Misinterpretation occurs because tacit assumptions were never formalized | Uneven visibility; obvious to insiders, opaque to outsiders | Knowle erosi |

# Detection Principles

Detecting scaffolding problems requires proactive strategies that pierce the invisibility of infrastructure before failures propagate into materials AI systems. Five core principles provide a practical framework. Principle 1: Dependency Mapping involves explicitly charting every external resource in the pipeline, from data repositories such as the Materials Project [6] and AFLOW [7] to software libraries like Matminer [8], revealing hidden assumptions that would otherwise remain invisible [1, 3]. Principle 2: Version Pinning requires documenting exact versions of libraries and database snapshots to detect dependency drift early, as minor updates in PyTorch or pymatgen can silently alter model outputs [23]. Principle 3: Access Monitoring tracks the ongoing availability and latency of critical components, flagging potential access loss before it disrupts workflows [10]. Principle 4: Reproduction Testing entails periodic re-execution of published pipelines in fresh computational environments to expose reproducibility collapse [2]. Principle 5: Knowledge Auditing systematically captures and verifies tacit assumptions held by team members before knowledge erosion occurs [13, 26]. Applied together, these principles convert scaffolding from an unexamined background into a monitorable layer of the system [3].

# Mitigation Principles

Mitigation demands that scaffolding be treated as a first-class design element rather than an implicit given. Six principles guide resilient practice. Principle 1: Explicit Scaffolding Documentation requires every paper and repository to list all infrastructure dependencies in detail, making the invisible visible [3]. Principle 2: Containerization through Docker or Singularity packages entire software environments, neutralizing dependency drift and ensuring portability across compute platforms [2]. Principle 3: Data Archiving mandates local copies of key datasets from sources like JARVIS or OQMD rather than sole reliance on live database links [10, 17]. Principle 4: Dependency Minimization encourages developers to strip unnecessary external libraries, reducing surface area for failure. Principle 5: Infrastructure Independence designs models capable of swapping data sources or software stacks without retraining from scratch, avoiding lock-in [12]. Principle 6: Knowledge Capture creates living documentation of tacit practices and community norms through annotated notebooks or wikis, countering erosion [1, 13, 27-29]. These principles collectively address the four mechanisms of failure and promote long-term robustness in materials AI.

# Relation to Other Failure Modes

The scaffolding problem is distinct yet intimately related to other recognized failure modes. It extends beyond technical debt [2], which focuses on internal code compromises and modeling shortcuts, whereas scaffolding failures originate in external socio-technical layers such as databases and institutional resources [3]. Scaffolding failure constitutes a primary underlying driver of the broader reproducibility crisis, because infrastructure decay and access loss directly prevent regeneration of results even when code appears identical [1]. It also fuels epistemic debt by concealing unexamined assumptions about data provenance, measurement standards, and resource permanence that shape what counts as valid knowledge [4, 5]. Unlike purely algorithmic or data-quality failures, scaffolding issues arise in the supporting strata and propagate upward undetected, making them especially difficult to diagnose without the typology presented here.

# Implications for Materials AI Practice

Materials AI practice must change to treat scaffolding explicitly. Authors should document every infrastructure dependency, supply containerized environments, and archive critical data locally rather than depending on potentially unstable links [6, 8]. Reviewers must verify scaffolding documentation and question results that rest on unstated infrastructure. The wider community needs to develop standardized infrastructure documentation templates, launch preservation initiatives for key resources such as the Materials Project and AFLOW, and secure dedicated funding for maintenance of scientific scaffolding [7, 10]. These shifts would reduce silent dependency failures and knowledge gap failures while enhancing the long-term reliability and trustworthiness of data-driven materials discovery.

# Conclusion

The scaffolding problem reveals that materials AI systems are fundamentally dependent on unstated scientific infrastructure that remains invisible until it fails. By defining this failure mode, delineating its six types of infrastructure, four mechanisms, and four archetypal modes, and by offering detection and mitigation principles, this analysis establishes scaffolding failure as a distinct and preventable concern. Recognizing scaffolding as essential rather than peripheral is critical for building trustworthy, reproducible materials AI capable of sustained contributions to materials innovation.


# Acknowledgements

None

# Conflict of interest

None

# Financial support

None


# Ethics statement

None

Received: 15 Jul 2023 Revised: 15 Sep 2023 Accepted: 31 Oct 2023
Published online: 18 January 2024



# References


De Sousa TC, Melo CD. Sustainable infrastructure, industrial ecology, and eco-innovation: Positive impact on society. In: Industry, innovation and infrastructure. Cham: Springer International Publishing; 2021. p. 1093-102.

Huyen C. Designing machine learning systems. Sebastopol: O'Reilly Media; 2022.

DeCost BL, Hattrick-Simpers JR, Trautt Z, Kusne AG, Campo E, Green ML. Scientific AI in materials science: A path to a sustainable and scalable paradigm. Mach Learn Sci Technol. 2020;1(3):033001.

Butler KT, Davies DW, Cartwright H, Isayev O, Walsh A. Machine learning for molecular and materials science. Nature. 2018;559(7715):547-55.

Schmidt J, Marques MR, Botti S, Marques MA. Recent advances and applications of machine learning in solid-state materials science. NPJ Comput Mater. 2019;5(1):83.

Stier SP, Kreisbeck C, Ihssen H, Popp MA, Hauch J, Malek K, et al. Materials acceleration platforms (MAPs): Accelerating materials research and development to meet urgent societal challenges. Adv Mater. 2024;36(45):2407791.

Carriço PJ. Computational materials screening using machine learning methods [master's thesis]. Coimbra: Universidade de Coimbra; 2021.

Ward L, Dunn A, Faghaninia A, Zimmermann NE, Bajaj S, Wang Q, et al. Matminer: An open source toolkit for materials data mining. Comput Mater Sci. 2018;152:60-9.

Zhao XG, Zhou K, Xing B, Zhao R, Luo S, Li T, et al. JAMIP: An artificial-intelligence aided data-driven infrastructure for computational materials informatics. Sci Bull. 2021;66(19):1973-85.

Choudhary K, Garrity KF, Reid AC, DeCost B, Biacchi AJ, Hight Walker AR, et al. The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design. NPJ Comput Mater. 2020;6(1):173.

Huber SP, Zoupanos S, Uhrin M, Talirz L, Kahle L, Häuselmann R, et al. Aiida 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. Sci Data. 2020;7(1):300.

Wang Z, Chen A, Tao K, Cai J, Han Y, Gao J, et al. Alphamat: A material informatics hub connecting data, features, models and applications. NPJ Comput Mater. 2023;9(1):130.

Xu P, Ji X, Li M, Lu W. Small data machine learning in materials science. NPJ Comput Mater. 2023;9(1):42.

Sbailò L, Fekete Á, Ghiringhelli LM, Scheffler M. The NOMAD artificial-intelligence toolkit: Turning materials-science data into knowledge and understanding. NPJ Comput Mater. 2022;8(1):250.

Hu J, Stefanov S, Song Y, Omee SS, Louis SY, Siriwardane EM, et al. Org: A materials informatics web app platform for materials discovery and survey of state-of-the-art. NPJ Comput Mater. 2022;8(1):65.

Hegde VI, Borg CK, Del Rosario Z, Kim Y, Hutchinson M, Antono E, et al. Quantifying uncertainty in high-throughput density functional theory: A comparison of aflow, materials project, and OQMD. Phys Rev Mater. 2023;7(5):053805.

Shen J, Griesemer SD, Gopakumar A, Baldassarri B, Saal JE, Aykol M, et al. Reflections on one million compounds in the open quantum materials database (OQMD). J Phys Mater. 2022;5(3):031001.

Menon S, Lysogorskiy Y, Knoll AL, Leimeroth N, Poul M, Qamar M, et al. From electrons to phase diagrams with machine learning potentials using pyiron based automated workflows. NPJ Comput Mater. 2024;10(1):261.

Aykol M. Materials discovery through artificial intelligence. Bull Am Phys Soc. 2020;65.

Chibani S, Coudert FX. Machine learning approaches for the prediction of materials properties. APL Mater. 2020;8(8):080701.

Takeda S, Kishimoto A, Hamada L, Nakano D, Smith JR. Foundation model for material science. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2023;37(13):15376-83.

Wang C, He S, Luo S, Huan Y, Wu S. Brainunit: Integrating physical units into high-performance ai-driven scientific computing. bioRxiv. 2024:2024-09.

Ramprasad R, Batra R, Pilania G, Mannodi-Kanakkithodi A, Kim C. Machine learning in materials informatics: Recent applications and prospects. NPJ Comput Mater. 2017;3(1):54.

Hill J, Mannodi-Kanakkithodi A, Ramprasad R, Meredig B. Materials data infrastructure and materials informatics. In: Computational materials system design. Cham: Springer International Publishing; 2018. p. 193-225.

Liu Y, Zhao T, Ju W, Shi S. Materials discovery and design using machine learning. J Mater. 2017;3(3):159-77.

Ma Y, Xu P, Li M, Ji X, Zhao W, Lu W. The mastery of details in the workflow of materials machine learning. NPJ Comput Mater. 2024;10(1):141.

Lei X, Kim E, Baibakova V, Sun S. Lessons in reproducibility: Insights from NLP studies in materials science. arXiv [Preprint]. 2023:arXiv:2307.15759.

Wines D, Gurunathan R, Garrity KF, DeCost B, Biacchi AJ, Tavazza F, et al. Recent progress in the JARVIS infrastructure for next-generation data-driven materials design. Appl Phys Rev. 2023;10(4):041302.

Reed PM, Hadjimichael A, Moss RH, Brelsford C, Burleyson CD, Cohen S, et al. Multisector dynamics: Advancing the science of complex adaptive human-Earth systems. Earths Future. 2022;10(3):e2021EF002621.