Concept Paper

# Architectures for Data-Aware LLMs: Models that Reason About Their Own Training Signal

Feng Chen *

Posted Date: 8 December 2025

doi: 10.20944/preprints202512.0566.v1

Keywords: data-aware language models; epistemic state and uncertainty; data provenance and coverage; active learning and data acquisition; domain shift and robustness doi:10.20944/preprints202512.0566.v1

*Concept Paper*

# Architectures for Data-Aware LLMs: Models that Reason About Their Own Training Signal

**Feng Chen**

University of Chinese Academy of Sciences, China; 1401761846@qq.com

## Abstract

Current large language models (LLMs) are fundamentally *data-blind*: they inherit structure, biases, and gaps from their training corpora, yet they lack any explicit representation of where their knowledge came from, how reliable it is, or which parts of the world they have effectively "seen". As a result, standard models struggle to answer basic epistemic questions such as: *How confident should I be on this query, given my training signal? Which domains or populations are under-represented in my experience? Which new documents, experiments, or user interactions would most efficiently reduce my uncertainty?* In this Perspective, we outline an emerging paradigm of data-aware LLMs that treat training data and learning history as first-class objects of computation. We propose architectural mechanisms for encoding data provenance, density, diversity, and conflict into persistent meta-representations that are accessible at inference time. These meta-layers enable models to expose calibrated uncertainty, surface data gaps, and condition their behavior on explicit epistemic state. We then discuss how data-aware models can drive active learning loops—proposing targeted data acquisitions, negotiating access with human and machine partners, and continuously updating their own meta-representations—to remain aligned with evolving domains and standards. Finally, we highlight applications to domain shift detection, robustness, and scientific discovery, and we analyze open challenges in privacy, governance, and standardization of data meta-layers. We argue that making models explicitly aware of their epistemic roots is a necessary next step toward trustworthy deployment in high-stakes scientific, industrial, and societal contexts.

**Keywords:** data-aware language models; epistemic state and uncertainty; data provenance and coverage; active learning and data acquisition; domain shift and robustness

## 1. The Data-Blindness of Current LLM Architectures

Modern large language models (LLMs) are built on unprecedented volumes of data, yet they are almost completely *blind* to that data at inference time. During pre-training, models ingest trillions of tokens scraped from the web, books, code, and increasingly, domain-specific corpora [1–3]. The details of this process—source distributions, time ranges, quality filters—strongly shape downstream behavior: which languages are handled well, which scientific subfields are covered, which social biases are amplified [4–7]. But once training is finished, this entire history is compressed into a homogeneous parameter soup. Standard architectures do not maintain any explicit, queryable record of *where* particular knowledge came from, *how dense or diverse* the evidence was, or *how controversial* or time-sensitive the underlying data might be.

This data-blindness shows up immediately when we ask seemingly simple epistemic questions. Today's models cannot reliably answer: *How confident should you be on this question, given the distribution of your training signal? Are you extrapolating far beyond your data support? Are there major subpopulations, time periods, or domains you have barely seen?* At best, LLMs approximate such self-assessment via shallow cues (e.g., output probabilities or "I'm not sure" templates), but they have no direct handle on the structure of their training data. In most deployments, any awareness of data provenance, coverage, and quality lives outside the model—in data sheets, documentation, and governance processes maintained by humans [4,8–10]—and is not available as an internal object of reasoning.

The consequences are visible in familiar failure modes. Hallucinations—confidently stated but unsupported claims—are partly a symptom of poor calibration and next-token objectives [11,12], but they also reflect structural data ignorance. When a model fabricates a nonexistent citation or invents numerical values, it cannot distinguish "I am interpolating within a dense, well-covered region of data" from "I have never really seen this before, but something like this *feels* plausible." Techniques like temperature scaling, ensembles, and post-hoc calibration operate entirely on the output distribution [13,14]; they do not give the model access to information such as "this domain was sparsely represented" or "my training data here consisted almost entirely of synthetic or low-quality sources." In scientific and industrial settings—e.g., designing interfacial-flow experiments, recommending process parameters for new materials, or summarizing rapidly evolving biomedical knowledge—this disconnect can be dangerous: models may sound equally confident in well-established regimes and in regions that are effectively extrapolations from a handful of noisy examples.

A related limitation is the inability to self-identify out-of-distribution (OOD) inputs. Foundation models are increasingly deployed in settings where the environment shifts over time—new regulations, novel materials, emerging pathogens, changing social norms [4,7,8,10]. There is a rich literature on OOD detection and domain shift in machine learning, often based on density estimation, feature-space distances, or specialized detectors [15–17]. But current LLMs do not carry a structured representation of the *regions* of data space they were trained on; instead, they rely on generic heuristics such as low token-level likelihood or unstable next-token predictions. Without an explicit notion of "this query lies far from my experience" grounded in training-data structure, models struggle to adapt their behavior—e.g., by deferring to human experts, requesting more context, or refusing to answer.

Recent audits of training data for large models further underscore how opaque this relationship is. Studies of web-scale corpora such as The Pile, Common Crawl derivatives, and proprietary mixtures show highly skewed coverage across languages, geographic regions, disciplines, and demographic groups [5–7,18]. For instance, U.S. and European sources dominate many datasets; low-resource languages and Global South media are under-represented; scientific and technical materials cluster around a few well-indexed publishers and preprint servers. LLMs trained on these distributions inevitably inherit these skews, but in today's architectures they cannot expose that fact. Users interacting with the model see only fluent text, not a contextualization like: "Most of my knowledge on this topic comes from English-language sources before 2022, with little coverage of practice in country X" [4,8].

The issue is particularly acute for domain-specific and industrial deployments. Scientific LLMs are increasingly fine-tuned on curated datasets of papers, lab protocols, simulation logs, and proprietary measurements [19–22]. Autonomous laboratory systems and "robotic scientists" can even generate their own experimental data, closing loops between models and instruments [20,23,24]. However, once these signals are folded into the weights, the model typically loses the ability to answer: *Which parts of this conclusion rest on public literature vs. internal lab data? Which regimes in my parameter space (e.g., certain ranges of droplet impact speeds, substrate patterns, or temperatures) have dense experimental coverage and which are extrapolations?* For high-stakes decisions—such as changing process conditions in an industrial pilot line or proposing a new experimental regime in interfacial fluid dynamics—stakeholders need precisely this kind of epistemic transparency.

One might argue that careful documentation and governance around datasets can compensate for architectural data-blindness. Datasheets for datasets, model cards, and similar documentation practices are important steps forward [4,8,10]. They describe sources, collection procedures, and known limitations, and they can be used to inform deployment policies. But they are static, external artifacts. They cannot adapt dynamically to the *actual queries* and interaction histories of a running model, nor can the model itself reference them as part of its internal reasoning. If a scientific assistant

LLM is asked about a newly proposed droplet-rotation experiment or a rarely studied polymer system, there is no built-in pathway for it to check, "Have I seen anything like this in my training data?" or "Is this domain under-represented in my experience?"—unless such checks are approximated via ad-hoc retrieval or manually wired metadata.

Moreover, as models evolve toward continual and online learning, the notion of a fixed training set becomes obsolete. Models updated with fresh data—new papers, lab results, user feedback—are effectively experiencing a *longitudinal* training history [7,19,23]. Without explicit meta-representations of this history, it becomes increasingly hard to answer questions like: "Which version of the model saw which experiments?", "Did this conclusion depend on early, superseded data?", or "Have we introduced conflicting evidence that the model has not reconciled?" These are precisely the questions that matter for scientific reproducibility and regulatory oversight in domains such as medicine, materials, and finance [8,9,21].

All of this suggests that simple output-level heuristics—confidence scores, "I am not sure" templates, or generic safety disclaimers—are not enough. What is missing is *architectural* support for data awareness: mechanisms that make training data and learning history a first-class object of computation. That means building internal structures that encode, at some level of abstraction, (i) data provenance (which sources, time periods, and pipelines contributed to particular knowledge), (ii) data density and diversity (how many and how varied the examples are in a region of task or feature space), and (iii) data conflict (where sources disagree or where evidence is mixed). Only with such structures can a model begin to reason explicitly about questions like "How strong is my evidence?", "Where are my blind spots?", and "What additional data would most improve my understanding?"

In this Perspective, we use the term data-blindness to describe the absence of such structures in current architectures. It is not that models encode *no* information about data—they clearly do, as evidenced by their performance—but that this information is locked into weights and activations in a form that is not designed for introspection or external querying. By contrast, data-aware LLMs would expose *meta-representations* of their training signal as part of their epistemic state, accessible both to themselves (for internal reasoning) and to external agents (for governance and oversight).

The rest of this article develops that idea. Section 2 introduces concrete architectural mechanisms for representing data provenance, density, and diversity inside models. Section 3 discusses how these meta-representations can be queried at inference time to support epistemic self-assessment, domain-shift detection, and calibrated explanation. Section 4 explores how data-aware models can drive active learning loops and scientific discovery by proposing high-value data acquisitions. Section 5 analyzes governance challenges: privacy, access control, and the need for standards around data meta-layers. Our central claim is that such architectures are not a luxury but a necessity if LLMs are to be trusted in high-stakes scientific and industrial roles where understanding *how* and *why* a model knows something is as important as what it predicts.

## 2. Representing Data Provenance, Density, and Diversity Inside Models

If data-blindness is the problem, then data meta-representations are the cure. The core idea of data-aware LLMs is that properties of the training signal—where it came from, how much of it exists, how diverse or conflicting it is—should not live only in offline spreadsheets and governance docs, but as *structured, queryable objects inside the model*. This section sketches architectural directions for representing three key aspects of training data within LLMs themselves: provenance, density, and diversity.

At a conceptual level, we can borrow from long-standing notions of data provenance and lineage in databases and enterprise systems: structured records describing the origin, movement, and transformation of data, often down to individual tables or files [25–27]. Similar ideas have been imported into ML tooling in the form of experiment-tracking systems and metadata stores that log which datasets, model versions, and preprocessing pipelines were used in each training run, enabling reproducibility and auditing [28]. In parallel, documentation frameworks such as datasheets, data cards, and model cards provide *human-readable* summaries of dataset composition, sources, and risks [8,9,29]. However, all of these live *outside* the model. Our goal here is to imagine a model-internal analogue: a “data meta-layer” that encodes, in a compact but structured way, the training signal’s provenance and coverage.

A natural starting point is to define data regions—coarse-grained partitions of the training data along dimensions that matter for downstream behavior. Regions might correspond to dataset shards (“web dump X vs. curated scientific corpus Y”), domains (“biomedicine vs. fluid mechanics vs. legal text”), time slices (“pre-2020 vs. post-2020”), or quality tiers (expert-curated vs. weakly labeled). During training, each example is tagged with one or more region identifiers. From the model’s perspective, these identifiers can be treated much like tasks or domains in multi-task learning: they condition routing, representation, and auxiliary objectives [32].

Architecturally, mixture-of-experts (MoE) and related conditional computation methods are a natural substrate for such region-aware modeling. MoE layers, as popularized by Shazeer *et al.* [30], activate only a small subset of “expert” subnetworks per input token, gating computation based on learned routing functions. In a data-aware variant, routing decisions could be informed not only by token content but also by explicit region tags. For example, some experts could specialize on different data regions (“web English news”, “Chinese scientific papers”, “internal lab notebooks on polymer synthesis”), while a gating network learns to route examples accordingly during pre-training and fine-tuning. The mapping from experts to regions does not need to be one-to-one, but even a soft association—“this expert tends to serve region R”—already provides a handle for linking parameters to provenance.

At a finer granularity, retrieval-augmented generation (RAG) architectures show how to maintain explicit, document-level provenance in *external* memory. RAG systems encode a corpus into a vector index, retrieve relevant passages at inference time, and condition generation on these passages, retaining clear pointers back to the underlying documents [31]. This external memory can carry rich metadata: URLs, timestamps, authors, licensing, or even experiment identifiers. Today, such metadata is typically used for filtering or display, not as a first-class object of reasoning. A data-aware architecture would go further: it would integrate *summaries* of these metadata fields into an internal meta-layer, so that the model can say, for instance, “my answer is based primarily on 2018–2020 arXiv preprints from subfield X, with little coverage of experiments in regime Y.”

One concrete design pattern is a dual representation:

- a standard parametric core (transformer + MoE, etc.) that learns task performance,
- plus an attached meta-representation over data regions, implemented as a small learned table or graph whose nodes correspond to regions and whose entries track statistics such as total tokens seen, number of distinct sources, distribution over time, and estimated label noise.

Training then becomes multi-headed: for each batch, we update not only the main weights via the usual loss, but also the meta-layer via auxiliary objectives that encourage correct association between examples and region embeddings. For example, a region classifier head can be trained to predict the region tag from intermediate activations; a reconstruction loss can ensure that region embeddings preserve key metadata such as time or domain.

Density can be represented in this meta-layer through simple but powerful summary statistics. Each region node maintains counts of examples, tokens, or gradient-norm contributions; these can be stored as moving averages or low-precision accumulators in a side table. Because memory budgets are finite, we cannot track everything at full resolution, but approximate sketches can provide enough signal to distinguish “well-covered” from “sparse” regions. Diversity can be captured via proxies such as vocabulary entropy, embedding dispersion, or the number of distinct sources (e.g., distinct journals, websites, or labs) contributing to a region.

Equally important is conflict and inconsistency. Real datasets often contain contradictory labels or competing scientific claims. Provenance-aware tools in databases already treat data provenance as essential for debugging and auditing: when inconsistencies are found, provenance helps attribute

doi:10.20944/preprints202512.0566.v1

them to sources and transformations [25–27]. Inside a model, we can mirror this by tracking, for each region, how often examples from that region contribute to *disagreements* during training—for instance, by logging high-loss examples, gradient conflicts with neighboring regions, or cases where fine-tuning on region R degrades performance on region S. A simple scalar "conflict score" per region (or per pair of regions) could already be a useful signal in later sections: high-conflict regions might warrant special treatment in uncertainty estimates or active data acquisition.

Another design axis concerns the resolution of provenance. Document-level provenance (individual papers, reports, or experimental logs) is ideal for transparency but too fine-grained to embed directly for trillions of tokens. Practical architectures will likely be hierarchical: documents grouped into sources (journals, repositories, labs), which in turn belong to domains and time slices. The meta-layer then focuses on these coarser groupings, while a retrieval index maintains fine-grained links. When a model produces an answer, it can query both layers: the retrieval index to surface concrete citations, and the meta-layer to summarize coverage ("most evidence comes from source family A, time window B, with low diversity in region C").

It is worth emphasizing the relationship between these internal structures and external documentation like datasheets and data cards. External artifacts excel at capturing qualitative rationales and human judgments—for example, why a dataset was collected, what populations it represents, or which known harms it might cause [8,9,29]. Internal meta-layers, by contrast, are quantitative and operational: they must be small, fast, and tightly integrated with training and inference. A healthy data-aware ecosystem will need both. One can imagine pipelines where external data cards are parsed into structured tags and priors for the internal meta-layer ("this dataset is high-quality but narrow; this one is broad but noisy"), and where the internal layer periodically exports updated statistics back to external governance tools ("region X has now seen N additional experiments; diversity in region Y remains low").

Data-aware representations also interact with task structure. Many LLMs are now trained in multi-task or multi-domain settings, where examples come with task identifiers, prompts, or adapters. From a data-aware standpoint, we can treat *tasks* as one dimension of the data-region space. For example, open-domain QA on web text, code completion on GitHub, and scientific summarization of preprints might all be separate regions. A single example can live at the intersection of multiple regions: "biomedical QA from curated clinical notes collected in 2023" combines domain, task, source type, and time. The meta-layer's job is to provide a compressed but navigable map of this multi-dimensional space.

Finally, there is the question of who or what consumes these representations. Even before we expose meta-data to users, the meta-layer is useful during training and deployment: it can guide routing (e.g., selecting experts or retrieval sources appropriate to a region), influence loss weighting (e.g., up-weighting sparse or high-value regions), and inform regularization (e.g., encouraging smoothness across neighboring regions but robustness in high-conflict ones). In this sense, data-aware representations are not just passive logs but control surfaces for training and deployment. Later sections will argue that both the model itself (via internal "epistemic queries") and external actors (users, auditors, other agents) should be able to query parts of the meta-layer.

In summary, representing data provenance, density, and diversity inside models requires a blend of ideas from provenance tracking, mixture-of-experts, and retrieval-augmented architectures [25,28,30,31]. External governance artifacts—datasheets, model cards, data cards—give us a vocabulary for what needs to be tracked; MoE and RAG give us mechanisms for linking examples and parameters to regions and sources. The next step is to expose these meta-representations at inference time, turning them into part of the model's epistemic state so that it can answer not only "what do I think?" but also "*why* do I think this, and how well is it supported by my training signal?"

## 3. Inference-Time Access to Meta-Data: Querying One's Own Epistemic State

A data-aware LLM is only useful if its meta-representations of the training signal can actually be *used* at inference time. In today's systems, epistemic information is largely implicit: token probabilities, logit margins, or ensemble disagreement are treated as rough surrogates for "how sure the model is," but the model itself has no structured language for asking, *Why do I believe this? How far am I extrapolating beyond my data?* Bayesian deep learning has long emphasized the importance of distinguishing epistemic from aleatoric uncertainty, and of surfacing calibrated confidence to downstream decision-makers [31–33]. Yet most current LLMs expose only a thin slice of this internal uncertainty to users and toolchains. The central claim of this section is that inference-time access to a rich epistemic interface—grounded in the meta-data described in Section 2—is the operational heart of data-aware architectures.

Classical work in Bayesian neural networks and Monte Carlo dropout shows how model uncertainty can be approximated by treating dropout as a variational posterior over weights, providing principled epistemic confidence intervals for predictions [32,33]. This research also clarified that epistemic uncertainty reflects a lack of knowledge that can, in principle, be reduced with more data, while aleatoric uncertainty reflects irreducible noise in the data-generating process [31]. Data-aware LLMs should inherit this distinction, but extend it: instead of a single scalar "uncertainty," the model should be able to decompose its epistemic state into dimensions like *data sparsity*, *label disagreement*, *source disagreement*, *temporal staleness*, and *domain mismatch*, all defined relative to the meta-layer that encodes provenance, density, diversity, and conflict in the training corpus (Section 2).

Recent work on LLM self-knowledge shows that large models can, to a surprising extent, learn to estimate whether they are likely to be correct on a given query, and to abstain gracefully when they are not. Kadavath *et al.* demonstrate that language models "mostly know what they don't know" when appropriately prompted and evaluated, providing a baseline for introspective calibration [34]. Kapoor *et al.* go further, arguing that such behavior is not automatic: prompting alone does not yield well-calibrated uncertainties, but modest fine-tuning on graded examples can produce robust uncertainty estimators with low computational overhead [35]. These methods, however, treat uncertainty as an emergent property of the forward pass, not as a structured query into a persistent record of the model's learning history. In a data-aware architecture, the same self-evaluation head that predicts answer correctness would also query the meta-layer: it could report *why* a question lies near the knowledge boundary (e.g., "few training documents, high source conflict, rapid concept drift"), not just *that* it does.

The emerging literature on knowledge boundaries offers a useful framing for such epistemic interfaces. Yin *et al.* introduce the knowledge boundary as the set of questions a model can answer reliably under some range of prompts, and propose optimization-based methods to probe these boundaries systematically [36]. Li *et al.* survey this field, classifying different types of knowledge and mapping how models fail when queries move outside their parametric comfort zone [37]. Complementary benchmarks such as UnknownBench examine how models express uncertainty when confronted with queries they are unlikely to have seen in training, highlighting systematic overconfidence on truly novel questions [38]. Data-aware LLMs should treat "knowledge boundary" not as a latent geometric property of weights, but as a queryable object: given a question, the model should be able to answer both "what do I think?" and "where does this question sit relative to the regions of task and data space I was trained on?"

A large body of recent work focuses on hallucination detection and uncertainty quantification for LLMs, providing building blocks for such epistemic interfaces. Farquhar *et al.* use semantic entropy over multiple generations to detect confabulations, showing that entropy in *meaning space* is more predictive of hallucinations than simple token-level entropy [39]. Kossen *et al.* introduce semantic-entropy probes that approximate this signal from a single generation's hidden states, dramatically reducing computational cost while preserving much of the predictive power [40]. Beigi *et al.* propose InternalInspector, which learns to map internal representations across layers to confidence scores, outperforming logit-based baselines and other internal-state methods on hallucination detection and calibration tasks [41]. In a data-aware model, these probes would be augmented by explicit features from the meta-layer: for example, semantic entropy could be conditioned on the local density and disagreement of training samples in the relevant data cell, and InternalInspector-style classifiers could be trained to incorporate provenance and temporal freshness signals alongside activations.

Beyond hallucination prediction, there is growing interest in whether LLMs can exhibit a form of functional introspection—reporting facts about their own internal states and training rather than just about the external world. Binder *et al.* show that models can improve at predicting their own future outputs and error patterns by training on introspective tasks, effectively learning a self-model [42]. Work on emergent introspective awareness in large language models suggests that frontier systems can, to a limited degree, distinguish internally injected "thoughts" from genuine activations and report on these interventions, though the behavior is fragile and highly context-dependent [43]. Seo *et al.* argue that much apparent self-awareness in hallucination prediction is actually driven by "question-side shortcuts" rather than genuine model-side introspection, and propose metrics to disentangle these effects [44]. These findings reinforce a key design principle for data-aware architectures: inference-time epistemic queries should be grounded in explicit, structured state—such as the data meta-layer—rather than being left as free-floating emergent capabilities.

One promising direction is to distinguish *answer-level* from *query-level* epistemic access. Answer-level uncertainty estimates how likely a specific generated answer is to be correct, given the model and context. Query-level uncertainty instead asks whether the model is in a position to answer the question *at all*, before any tokens are generated. Chen *et al.* formalize query-level uncertainty and propose internal-confidence methods that estimate whether a query falls inside the model's knowledge boundary using pre-generation signals [45]. In a data-aware model, query-level checks would be implemented as fast lookups against the meta-layer: given a representation of the query in task/capability space, the system would retrieve approximate training density, domain coverage, and conflict statistics for nearby cells, and then decide whether to proceed, to call external tools (e.g., retrieval, simulators), or to abstain. Answer-level probes—semantic entropy, internal-state classifiers, self-evaluation heads—would then provide finer-grained epistemic signals conditional on a candidate answer [35,39–41].

From an architectural perspective, this suggests that every forward pass should be augmented with a parallel *epistemic channel*. For each query, the model would output not only a sequence of tokens, but also a structured epistemic report with fields such as: (i) confidence over correctness; (ii) distances to known capability clusters; (iii) data coverage metrics (e.g., number and diversity of supporting training cells); (iv) indicators of distribution shift (e.g., novelty in vocabulary, style, or source domains); and (v) recommended actions (trust, defer to retrieval, escalate to human, or abstain). This channel need not expose raw training data—indeed, for privacy and IP reasons it often cannot—but it can expose *aggregated* statistics computed over the meta-layer. Benchmarks targeting knowledge boundaries and uncertainty-sensitive QA can then be used to measure how well these epistemic signals track actual accuracy, robustness, and generalization performance [36–38].

Finally, inference-time epistemic access is not just a UX nicety; it is a control signal for larger agentic systems. Confidence-aware chains of thought, multi-agent debate, and tool-use pipelines all rely on some notion of "when to think harder, when to call a tool, and when to stop." Recent studies of confidence in LLM reasoning show that self-reported or behavior-derived confidence correlates with correctness, but is often miscalibrated and task-dependent [34,35]. Combining these behavioral measures with meta-data–driven epistemic signals could enable more reliable decision policies: an orchestrating agent might, for instance, insist on external verification whenever the inferred data coverage is sparse or source disagreement is high, even if the base model's self-reported confidence is strong. In high-stakes scientific and industrial workflows, such policies may make the difference between a system that merely "sounds right" and one whose outputs are grounded in a transparent, queryable understanding of what it knows—and how it knows it.

## 4. Active Data Acquisition and Applications to Robustness and Scientific Discovery

Once models can represent and query properties of their own training signal, the natural next step is to *use* that epistemic state to drive active data acquisition. Rather than passively consuming whatever corpora are available, data-aware LLMs can participate in closed-loop systems that ask: *Where is my knowledge thin, biased, or conflicting? Which new data points would most improve my understanding or robustness?* This is the core intuition behind decades of work on active learning and Bayesian experimental design, but now applied at the scale and generality of foundation models.

Classical active learning studies the setting where labels (or experiments) are expensive and unlabeled data (or candidate experiments) are plentiful. The learner iteratively selects the most informative points to query, often based on uncertainty, expected error reduction, or version-space criteria [46]. In continuous design spaces, Bayesian optimization (BO) formalizes this process as optimizing an expensive black-box function with a probabilistic surrogate, typically a Gaussian process or Bayesian neural network, and an acquisition function that trades off exploration and exploitation [47–49]. BO has proved especially powerful for experiment planning in chemistry and materials, where each experiment may take hours or days and the design space is high-dimensional and constrained [49–51].

Data-aware LLMs provide an opportunity to "lift" these ideas from low-level parameter spaces (e.g., reaction temperatures, compositions) to semantic task and data spaces. Instead of merely asking, "which point in this physical parameter space should we sample next?", a data-aware model can ask, "which *data region* in my meta-layer—defined by domain, time, population, or capability—would most reduce my epistemic uncertainty on questions like the one I just saw?" In other words, the *actions* in active learning become not just individual experiments, but targeted data acquisitions: reading specific literatures, requesting domain experts to annotate edge cases, or suggesting new experiments for a self-driving lab.

Self-driving laboratories (SDLs) and autonomous experimentation systems already embody this closed-loop ideal at the level of physical experiments: robotic platforms coupled to ML decision engines that iteratively select experiments, run them, measure outcomes, and update models [51–53]. SDLs have been demonstrated for reaction optimization, thin-film deposition, alloy discovery, and other areas, often achieving order-of-magnitude reductions in the number of experiments needed compared to grid searches [50–53]. Kusne *et al.*'s CAMEO platform, for example, uses Bayesian active learning to control synchrotron X-ray diffraction measurements and discover new phase-change materials with roughly tenfold fewer experiments than manual baselines [50]. Stach *et al.* and Tom *et al.* review the broader ecosystem of autonomous experimentation, emphasizing that adaptive sampling—changing where you measure based on what you have already seen—is central to realizing SDLs' promise [52,53].

What data-aware LLMs add is an explicit epistemic map over the data feeding these loops. In a conventional SDL, the active learner evaluates uncertainty in a local surrogate model defined over a specific experimental task (e.g., composition–property mapping). The rest of the lab's knowledge—papers, historical data, simulations—is either folded implicitly into that surrogate or handled via ad-hoc retrieval. In a data-aware architecture, the LLM that helps design experiments would consult its meta-layer before proposing new measurements: if the data region corresponding to, say, high Weber-number droplet impacts on structured substrates is sparsely populated, while low-Weber regimes are dense and consistent, the model can recommend experiments that specifically probe the high-Weber regime or fill gaps in the parameter–structure grid. For a materials synthesis campaign, it might notice that certain classes of ligands or processing temperatures are underrepresented in the combined literature+lab data and propose a BO loop that systematically explores those under-sampled combinations.

This suggests a hierarchy of active learning loops. At the outer level, the data-aware LLM reasons over its meta-layer and interaction logs to identify *which data regions to prioritize*: perhaps recent user queries indicate growing demand for advice on a new polymer family, or for regulatory changes in a particular jurisdiction. It can then propose, to a human or an automated pipeline, a plan to acquire targeted data: scraping new corpora, commissioning domain experts to write tutorials, or allocating SDL time to measure missing regimes in the relevant experiments. At the inner level, within each region, conventional active learning and BO methods select specific instances or experiments, guided by uncertainty and constraints [46–51].

Robustness is another natural beneficiary of this data-aware active acquisition. Distribution shift and group disparities are well-known challenges in ML; distributionally robust optimization (DRO) frameworks seek decisions that perform well under worst-case perturbations within an ambiguity set of distributions [56–58]. While DRO has attractive theoretical guarantees, in practice its effectiveness hinges on how the ambiguity set is chosen and how it relates to actual data geometry and deployment scenarios. A data-aware LLM, with its explicit decomposition of the training signal into regions (e.g., demographic groups, domains, time slices), can instantiate DRO-like objectives at the *data-region level*: amplifying loss contributions from high-risk or under-served regions, or actively querying new data to shrink ambiguity sets where performance is most fragile.

In other words, instead of treating robustness as a static property of a finished model, we can view it as a continual data acquisition policy: monitor which regions of capability and data space contribute most to worst-case errors, then allocate data collection and fine-tuning resources accordingly. For example, if a scientific assistant model consistently underperforms on questions about emerging energy materials or on safety-critical aspects of wet-lab protocols, epistemic metrics from the meta-layer can pinpoint these regions. The system can then respond by: (i) surfacing these weaknesses to human overseers; (ii) recommending that certain kinds of queries be deferred or double-checked; and (iii) proposing concrete data acquisitions—such as new benchmark tasks, curated corpora, or targeted SDL campaigns—to improve performance in those regions.

Scientific discovery provides especially rich ground for such epistemically guided exploration. Traditional BO-based experiment planning already accelerates discovery by focusing on promising regions of parameter space [47–51,59]. But data-aware LLMs can go further by considering not just the local surrogate for a single objective, but the *global structure* of the model's knowledge. For instance, in interfacial fluid dynamics, a data-aware assistant might notice that many experiments exist for droplet impact on smooth, homogeneous substrates, while there is a paucity of data on anisotropic or fractal surfaces. It can then propose a research program—potentially executed by a SDL—specifically designed to fill those structural gaps, guided by priors about where rich phenomena (e.g., self-rotation, capillary instabilities, Kelvin–Raoult–Stefan coupling) are likely to emerge.

Similarly, in materials discovery, self-driving labs already combine high-throughput synthesis and characterization with ML planners to explore composition–structure–property landscapes [51–53,60]. A data-aware LLM sitting "above" these platforms could integrate information across multiple SDLs and modalities—linking simulations, experiments, and literature—to identify global blind spots: classes of compounds, processing routes, or environmental conditions that are systemically underexplored. Active proposals might range from "run BO over this new alloy composition range on SDL A" to "launch a data-collection effort for long-term stability of these polymers in humid environments," all justified by explicit meta-layer metrics quantifying coverage and uncertainty.

Of course, turning these ideas into practice raises challenges. Active learning literature is full of cautions: naive uncertainty sampling can focus on outliers and label noise; acquisition functions can get stuck in local regions; sample efficiency gains depend heavily on model mis-specification and initial coverage [46–49]. Data-aware LLMs must therefore combine epistemic signals from the meta-layer with robust acquisition strategies, potentially leveraging ensembles, model-switching, or hybrid exploration policies to avoid pathologies. In autonomous experimentation, additional constraints—safety limits, resource budgets, and hardware constraints—must be accounted for explicitly, as emphasized in chemistry-focused BO frameworks with known design constraints [49–51].

Despite these hurdles, the payoff is substantial. By closing the loop between epistemic self-knowledge and data acquisition, data-aware LLMs move us toward scientific and industrial systems that do not merely answer questions, but also *decide which questions to ask next, and which measurements or documents to seek*, in a principled, transparent way. For high-stakes applications—from optimizing nucleic-acid extraction materials to exploring new droplet-driven microfluidic paradigms—this shift could transform LLMs from static advisors trained once on frozen corpora into active partners in an ongoing, data-driven scientific process.

## 5. Governance of Data Meta-Layers: Privacy, Standards, and Long-Term Challenges

If data-aware LLMs make training data and learning history a first-class object of computation, they also make it a first-class object of governance. The very same meta-layers that enable better epistemic calibration, active learning, and scientific discovery (Sections 2–4) encode sensitive information about *who* contributed which data, *when* and *how* it was used, and *where* knowledge gaps lie. This is invaluable for transparency and auditing [4,7–10], but it raises serious questions about privacy, intellectual property, security, and long-term stewardship.

A first concern is privacy and re-identification risk. Even when raw training data are not directly exposed, meta-layers that summarize provenance, density, and coverage may leak information about individuals or organizations. For example, a data-aware LLM deployed in healthcare might maintain counts and statistics over data regions corresponding to specific hospitals or patient cohorts. If these statistics are exposed too granularly—say, via per-hospital coverage metrics or shift reports that can be queried by arbitrary users—an attacker could combine them with external side information to infer sensitive properties of individuals or small groups, echoing classic re-identification attacks on "anonymized" datasets [8,9,61,62]. Differential privacy was developed precisely to limit such inferences by bounding how much any single record can influence observable outputs [61], but most current LLM training practices do not use strong differential privacy, and even when they do, it is unclear how DP guarantees propagate to *derived* meta-layers.

This suggests that data meta-layers must be designed with access control and abstraction in mind. Internally, fine-grained provenance and density statistics may be necessary for robust epistemic reasoning and active learning (Sections 3–4). Externally, only *coarsened* or *aggregated* views should be exposed: for example, reporting coverage at the level of broad domains or time windows, not individual organizations or small demographic groups. Access can also be tiered: developers and auditors might see more detailed diagnostics under confidentiality agreements, while end users get high-level summaries ("this topic is under-represented in my training data; treat this answer with caution"). In high-stakes domains, regulators may require that internal meta-layers be auditable under secure conditions—analogous to how safety cases and incident logs are examined in other safety-critical industries—without being made public.

A second set of issues concerns ownership and intellectual property. Meta-layers make visible which sources and regions contribute to model behavior: that a particular pharmaceutical company's trial data or a specific lab's droplet-impact experiments [19–24] underpin certain capabilities, or that a proprietary corpus dominates coverage in some regime. This is welcome from the standpoint of scientific credit and data governance [4,7–10], but it complicates IP relationships: if a model's meta-layer reveals that a capability relies heavily on a given partner's data, does that partner gain new rights (or liabilities) with respect to that capability? How should licensing terms distinguish between training on raw data and *deriving* high-level coverage summaries? And how can organizations share meta-layers with each other (e.g., to discover data gaps or overlaps) without revealing commercially sensitive details of their holdings? These questions echo ongoing debates around data cards, datasheets, and model cards [8,9,29], but add a new twist: the artifacts are no longer static documents but *live, queryable components* of deployed systems.

Third, realizing the full value of data-aware architectures will require standardization and interoperability. Today, each lab or company that documents its datasets or models tends to define its own ontology: ad-hoc labels for domains, tasks, populations, and quality tiers. For meta-layers to be comparable and composable—for example, to understand how two models' coverage overlaps or to merge meta-information from multiple organizations—there must be shared schemas for key concepts: what counts as a "data region," which dimensions are tracked (domain, time, geography, demographic attributes, experimental conditions), and how statistics are computed and updated [7–10,25–29]. This is analogous to existing standardization efforts in scientific data infrastructures, where communities have developed domain ontologies and metadata standards to enable cross-dataset search and integration. For scientific LLMs, such standards might build on existing ontologies in materials, biology, and fluid mechanics, tying meta-layers directly to domain concepts like composition, process conditions, or flow regime.

Standardization has a governance aspect as well. Regulators and professional bodies are beginning to demand more structured documentation of AI systems' training and evaluation data; the European AI Act, for example, includes obligations around data quality and documentation for high-risk systems. While our discussion is not tied to any specific law, data meta-layers could become a key way to satisfy such requirements, providing machine-readable evidence about coverage, sources, and known gaps [7–10]. Conversely, if meta-layers are poorly specified or left entirely proprietary, they may become a new vector for opacity: systems could claim "data awareness" without offering any verifiable structure or semantics. Governance frameworks will need to specify not just that meta-layers exist, but what *minimum information* they must encode and how that information should be audited.

Even with careful design, data-aware architectures open up new failure modes. Models may "hallucinate" meta-data just as they hallucinate answers: if epistemic reports are generated by the same language interface as ordinary text, users may be misled into believing that a model's training coverage is broader or more balanced than it actually is. Internal meta-layers themselves may be biased or stale if they are not kept in sync with evolving training pipelines and data sources. For instance, if new experiments on condensation and wetting are added to a lab's data lake [19–24] but the coverage statistics for the corresponding region are not updated, the model might continue to act as if that region were under-sampled, mis-prioritizing active learning proposals (Section 4). More subtly, if meta-layers are used in governance (e.g., to demonstrate compliance with fairness or safety requirements), organizations may have incentives to "optimize" these summaries, under-reporting gaps or over-smoothing conflicts [8,9,56–58].

Addressing these risks will require independent auditing and red-teaming of meta-layers, not just of model outputs. Traditional algorithmic auditing and bias assessment already emphasize the need for external scrutiny of datasets and model behavior [8–10,63,64]. For data-aware LLMs, auditors will need tools to probe the consistency between meta-layers and reality: sampling raw training data to verify coverage statistics, injecting known synthetic data regions to test whether they are reflected properly in the meta-layer, and designing adversarial probes that try to elicit incorrect or deceptive epistemic reports. In scientific settings, this could include cross-checking meta-layer claims about the density of experiments or simulations in certain regimes (e.g., droplet impact at particular Weber and Reynolds numbers) against lab notebooks and data repositories.

Finally, there is the question of long-term stewardship. Large models and their meta-layers will be updated, fine-tuned, merged, and distilled over time [2,3,12,13]. Each such operation transforms not only the model's capabilities but also its epistemic roots. If we merge two models trained on different scientific corpora, how should their meta-layers be combined? If we distill a large expert ensemble into a smaller student model, how do we summarize the student's *effective* training signal and associated uncertainties? If we deprecate certain datasets for ethical or scientific reasons (e.g., flawed experiments, retracted papers [7–10,18]), how do we reflect that in both the model and its meta-layer? These questions are not new in principle—version control and deprecation are familiar in software and data management—but data-aware LLMs raise the stakes by tying them directly to a system's epistemic self-understanding and its obligations to users and regulators.

In sum, data meta-layers are double-edged: they are the key enabler for epistemically transparent, actively learning, scientifically useful LLMs, but they also create new surfaces for privacy leakage, misaligned incentives, and governance failure. The same care that the community has invested in dataset documentation, fairness audits, and safety evaluation [4,7–10,63,64] will have to be extended to these meta-structures. If we succeed, data-aware architectures could become the backbone of *epistemically accountable* AI: systems that can not only answer questions, but also explain and justify the structure of their own training signal, in ways that align with scientific norms and societal expectations.

## References

1. Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.
2. Kaplan, Jared, et al. "Scaling laws for neural language models." arXiv preprint arXiv:2001.08361 (2020).
3. Hoffmann, Jordan, et al. "Training compute-optimal large language models." arXiv preprint arXiv:2203.15556 (2022).
4. Bender, Emily M., et al. "On the dangers of stochastic parrots: Can language models be too big?." Proceedings of the 2021 ACM conference on fairness, accountability, and transparency. 2021.
5. Gao, Leo, et al. "The pile: An 800gb dataset of diverse text for language modeling." arXiv preprint arXiv:2101.00027 (2020).
6. Dodge, Jesse, et al. "Documenting large webtext corpora: A case study on the colossal clean crawled corpus." arXiv preprint arXiv:2104.08758 (2021).
7. Wiggins, Walter F., and Ali S. Tejani. "On the opportunities and risks of foundation models for natural language processing in radiology." Radiology: Artificial Intelligence 4.4 (2022): e220119.
8. Mitchell, Margaret, et al. "Model cards for model reporting." Proceedings of the conference on fairness, accountability, and transparency. 2019.
9. Gebru, Timnit, et al. "Datasheets for datasets." Communications of the ACM 64.12 (2021): 86-92.
10. Liang, Percy, et al. "Holistic evaluation of language models." arXiv preprint arXiv:2211.09110 (2022).
11. Ji, Ziwei, et al. "Survey of hallucination in natural language generation." ACM computing surveys 55.12 (2023): 1-38.
12. Kalai, Adam Tauman, et al. "Why language models hallucinate." arXiv preprint arXiv:2509.04664 (2025).
13. Guo, Chuan, et al. "On calibration of modern neural networks." International conference on machine learning. PMLR, 2017.
14. Ashukha, Arsenii, et al. "Pitfalls of in-domain uncertainty estimation and ensembling in deep learning." arXiv preprint arXiv:2002.06470 (2020).
15. Hendrycks, Dan, and Kevin Gimpel. "A baseline for detecting misclassified and out-of-distribution examples in neural networks." arXiv preprint arXiv:1610.02136 (2016).
16. Ren, Jie, et al. "Likelihood ratios for out-of-distribution detection." Advances in neural information processing systems 32 (2019).
17. Hu, Jun, and Zhan-Long Wang. "Dynamic Wetting and Spreading of High-Viscosity Liquids on Grooved Substrates." (2025).
18. Hu, Jun, and Zhan-Long Wang. "The effect of hygroscopic liquids on the spatial controlling of condensation on low-temperature surfaces." Surfaces and Interfaces 55 (2024): 105430.
19. Bran, Andres M., et al. "Chemcrow: Augmenting large-language models with chemistry tools." arXiv preprint arXiv:2304.05376 (2023).
20. Häse, Florian, Loïc M. Roch, and Alán Aspuru-Guzik. "Next-generation experimentation with self-driving laboratories." Trends in Chemistry 1.3 (2019): 282-291.
21. Fries, Jason A., et al. "Weakly supervised classification of aortic valve malformations using unlabeled cardiac MRI sequences." Nature communications 10.1 (2019): 3111.
22. Hu, Jun, and Zhan-Long Wang. "Analysis of fluid flow in fractal microfluidic channels." arXiv preprint arXiv:2409.12845 (2024).

23. Hu, Jun, et al. "The effect of substrate temperature on the dry zone generated by the vapor sink effect." Physics of Fluids 36.6 (2024).
24. Hu, Jun, and Zhan-Long Wang. "Crystallization morphology and self-assembly of polyacrylamide solutions during evaporation." arXiv preprint arXiv:2403.20191 (2024).
25. Hu, Jun, and Zhan-Long Wang. "Inhibition of water vapor condensation by dipropylene glycol droplets on hydrophobic surfaces via vapor sink strategy." arXiv preprint arXiv:2311.03930 (2023).
26. Bose, Rajendra, and James Frew. "Lineage retrieval for scientific data processing: a survey." ACM Computing Surveys (CSUR) 37.1 (2005): 1-28.
27. Buneman, Peter, Sanjeev Khanna, and Tan Wang-Chiew. "Why and where: A characterization of data provenance." International conference on database theory. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.
28. Wang, Zhan-Long, et al. "Suppression of water vapor condensation by glycerol droplets on hydrophobic surfaces." arXiv preprint arXiv:2311.03068 (2023).
29. Xu, Yanchao, et al. "Facet-dependent electrochemical behavior of au–pd core@ shell nanorods for enhanced hydrogen peroxide sensing." ACS Applied Nano Materials 6.20 (2023): 18739-18747.
30. Shazeer, Noam, et al. "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer." arXiv preprint arXiv:1701.06538 (2017).
31. Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in neural information processing systems 33 (2020): 9459-9474.
32. Wang, Zhan-Long, and Kui Lin. "The multi-lobed rotation of droplets induced by interfacial reactions." Physics of Fluids 35.2 (2023).
33. Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." international conference on machine learning. PMLR, 2016.
34. Wang, Zhanlong, et al. "Spontaneous motion and rotation of acid droplets on the surface of a liquid metal." Langmuir 37.14 (2021): 4370-4379.
35. Wang, Zhanlong, et al. "Realization of self-rotating droplets based on liquid metal." Advanced Materials Interfaces 8.3 (2021): 2001756.
36. Yin, Xunjian, et al. "Benchmarking knowledge boundary for large language models: A different perspective on model evaluation." arXiv preprint arXiv:2402.11493 (2024).
37. Li, Moxin, et al. "Knowledge boundary of large language models: A survey." Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2025.
38. Liu, Genglin, et al. "Examining LLMs' Uncertainty Expression Towards Questions Outside Parametric Knowledge." arXiv preprint arXiv:2311.09731 (2023).
39. Farquhar, Sebastian, et al. "Detecting hallucinations in large language models using semantic entropy." Nature 630.8017 (2024): 625-630.
40. Wang, Zhanlong, Kui Lin, and Ya-Pu Zhao. "The effect of sharp solid edges on the droplet wettability." Journal of colloid and interface science 552 (2019): 563-571.
41. Zhao, Ya-Pu, and Zhanlong Wang. "Moving Contact Line of Droplets on Structured Surfaces: Some Problems Relevant to Tribology." Surfactants in Tribology, Volume 6 (2019): 73-111.
42. Wang, ZhanLong, EnHui Chen, and YaPu Zhao. "The effect of surface anisotropy on contact angles and the characterization of elliptical cap droplets." Science China Technological Sciences 61.2 (2018): 309-316.
43. Chen, Sirui, et al. "From imitation to introspection: Probing self-consciousness in language models." Findings of the Association for Computational Linguistics: ACL 2025. 2025.
44. Seo, Yeongbin, Dongha Lee, and Jinyoung Yeo. "Quantifying Self-Awareness of Knowledge in Large Language Models." arXiv preprint arXiv:2509.15339 (2025).
45. Chen, Lihu, et al. "Query-level uncertainty in large language models." arXiv preprint arXiv:2506.09669 (2025).
46. Wang, Zhanlong, and Ya-Pu Zhao. "Wetting and electrowetting on corrugated substrates." Physics of Fluids 29.6 (2017).
47. Frazier, Peter I. "A tutorial on Bayesian optimization." arXiv preprint arXiv:1807.02811 (2018).

48. Shahriari, Bobak. Practical Bayesian optimization with application to tuning machine learning algorithms. The University of British Columbia (Canada), 2016.
49. Griffiths, Ryan-Rhys, and José Miguel Hernández-Lobato. "Constrained Bayesian optimization for automatic chemical design using variational autoencoders." Chemical science 11.2 (2020): 577-586.
50. Hickman, Riley J., et al. "Bayesian optimization with known experimental and design constraints for chemistry applications." Digital Discovery 1.5 (2022): 732-744.
51. Häse, Florian, Loïc M. Roch, and Alán Aspuru-Guzik. "Next-generation experimentation with self-driving laboratories." Trends in Chemistry 1.3 (2019): 282-291.
52. Stach, Eric, et al. "Autonomous experimentation systems for materials development: A community perspective." Matter 4.9 (2021): 2702-2726.
53. Tom, Gary, et al. "Self-driving laboratories for chemistry and materials science." Chemical Reviews 124.16 (2024): 9633-9732.
54. Bennett, Jeffrey A., and Milad Abolhasani. "Autonomous chemical science and engineering enabled by self-driving laboratories." Current Opinion in Chemical Engineering 36 (2022): 100831.
55. Ishizuki, Naoya, Ryota Shimizu, and Taro Hitosugi. "Autonomous experimental systems in materials science." Science and Technology of Advanced Materials: Methods 3.1 (2023): 2197519.
56. Rahimian, Hamed, and Sanjay Mehrotra. "Distributionally robust optimization: A review." arXiv preprint arXiv:1908.05659 (2019).
57. Staib, Matthew, and Stefanie Jegelka. "Distributionally robust optimization and generalization in kernel methods." Advances in Neural Information Processing Systems 32 (2019).
58. Słowik, Agnieszka, and Léon Bottou. "On distributionally robust optimization and data rebalancing." International Conference on Artificial Intelligence and Statistics. PMLR, 2022.
59. Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams. "Practical bayesian optimization of machine learning algorithms." Advances in neural information processing systems 25 (2012).
60. Sanchez-Lengeling, Benjamin, and Alán Aspuru-Guzik. "Inverse molecular design using machine learning: Generative models for matter engineering." Science 361.6400 (2018): 360-365.
61. Dwork, Cynthia, and Aaron Roth. "The algorithmic foundations of differential privacy." Foundations and trends® in theoretical computer science 9.3–4 (2014): 211-407.
62. Rocher, Luc, Julien M. Hendrickx, and Yves-Alexandre De Montjoye. "Estimating the success of re-identifications in incomplete datasets using generative models." Nature communications 10.1 (2019): 3069.
63. Raji, Inioluwa Deborah, and Joy Buolamwini. "Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products." Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. 2019.
64. Floridi, Luciano, and Josh Cowls. "A unified framework of five principles for AI in society." Machine learning and the city: Applications in architecture and urban design (2022): 535-545.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.