Computers & Graphics (2026)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

LLMVisExplorer: Visually Exploring the Knowledge Landscape of Large Language Models

ARTICLE INFO

Article history:
Received 30th September 2026

Large Language Models,
Hierarchical Visualization,
Explainable Artificial Intelligence,
Focus+context techniques

ABSTRACT

The performance of large language models (LLMs) in specific knowledge domains is often evaluated based on question and answer (Q&A) benchmark datasets. Although benchmark reports sometimes include domain-level breakdowns, evaluation results are still frequently summarized through aggregate accuracy values or coarse tables that make it difficult to inspect multi-level, multi-metric, and question-level performance variation. This is problematic because Q&A benchmark datasets are by design diverse and span broad domains, such as science (SciQ) or biomedicine (PubMedQA). We argue that overly coarse evaluation summaries can collapse meaningful details: strong performance in easier or overrepresented areas can overshadow weaknesses in critical or complex subdomains, making such summaries insufficient for downstream use. To enable better understanding of real-world model capabilities essential for downstream applications, we propose LLMVisExplorer, an interactive visualization system that facilitates in-depth evaluation of LLM performance within the subdomains of complex knowledge domains. Based on LLM evaluation design requirements grounded in the natural language processing (NLP) literature, LLMVisExplorer bridges the gap between coarse-grained benchmark reporting and actionable model insights by enabling multifaceted, visually guided performance analysis across hierarchical knowledge domains. It empowers both, LLM developers and LLM users, to identify domain-specific strengths and weaknesses, in order to inform model development, red-teaming, and deployment decisions. We discuss LLMVisExplorer's design requirements as well as its realization, and showcase its potential on several use cases evaluating single models as well as comparing multiple models, such as Claude 3.5 Sonnet, GPT-4o, and Yi-large. LLMVisExplorer's source code and data will be provided upon acceptance.

CC BY-NC 4.0

1. Introduction

The advent of Large Language Models (LLMs) has been a significant breakthrough in the field of natural language processing (NLP) and has shown impressive capabilities in various tasks, such as language modeling, sentiment analysis, question-answering, and machine translation [7]. LLMs are trained on massive text datasets that contain vast amounts of language data, which makes them perform particularly well on tasks that require contextual understanding and the ability to generate coherent and meaningful responses. Researchers were even able to show that state-of-the-art LLMs pass law exams [11] or even the United States Medical Licensing Examination (USMLE) [31].

To formally evaluate the knowledge capabilities of LLMs in large and diverse knowledge domains, Q&A benchmark datasets have become a widely adopted standard [57]. Prominent examples include the general MMLU [55], BioASQ in biomedicine [52] and GPQA in the sciences [35], all designed to cover a broad spectrum of questions with ground truth answers from different subdomains. While these datasets offer valuable benchmarks, we argue that the common practice of reporting a single aggregated accuracy score on these, fails to reflect the full complexity of model behavior. A model might achieve high overall accuracy by performing well on overrepresented or easier

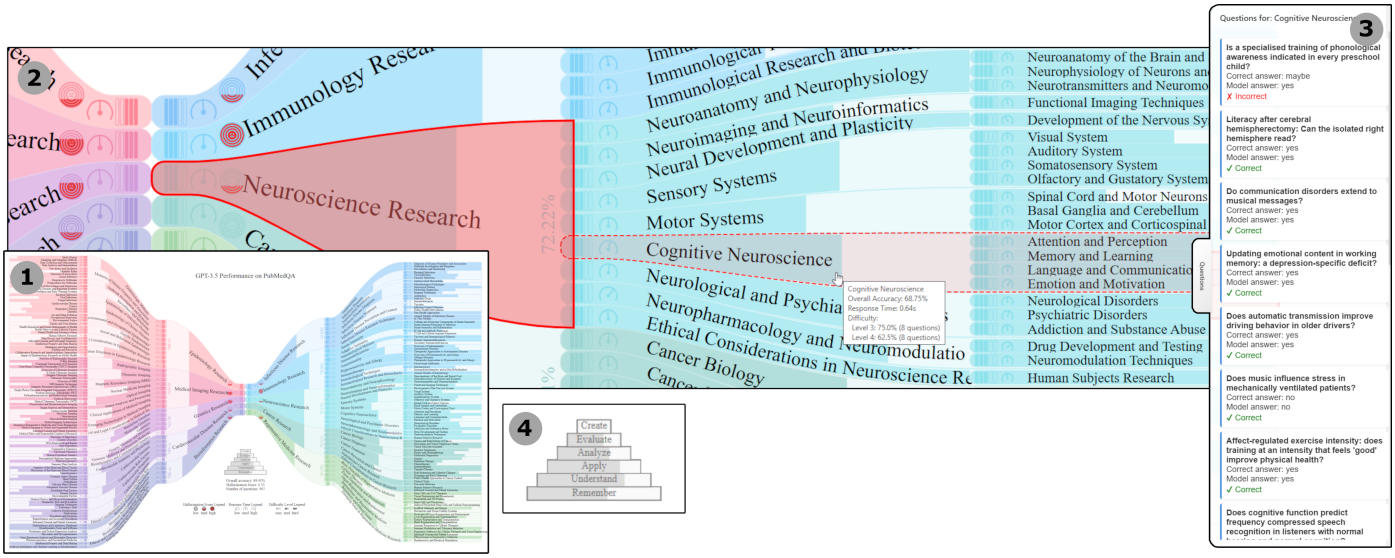


Fig. 1: LLMVisExplorer enables the exploration of LLM performance on Q&A benchmark datasets. Here GPT-3.5's performance on PubMedQA is broken down across biomedical subdomains. For each subdomaim, bars show accuracy, dots indicate question density, and icons encode difficulty level, response time and hallucination score in an overview visualization (1). Users can obtain details on demand, by hovering and selecting subdomains (2). Upon selection relevant questions are shown per subdomain (3). Grouping based on cognitive skill levels further supports a multifaceted, trustworthy view of GPT-3.5's strengths and limitations.

questions, while systematically failing in underrepresented but critical subdomains, such as rare diseases or newly emerging fields. This collapse of performance variation can mislead both developers and users - masking knowledge gaps, inflating trust, and ultimately posing risks in real-world applications [59]. Thus, without a multifaceted, subdomain-aware evaluation, it is difficult to obtain meaningful insights into a model's true strengths and limitations, which are essential for responsible deployment and further improvement [7].

To support a multifaceted evaluation of LLMs beyond single-number accuracies, we propose LLMVisExplorer, an interactive visualization system that enables both developers and practitioners to explore and compare model performance across the subdomains of one or more Q&A datasets (see Fig. 1). We do not view visualization as a replacement for hierarchical metric reporting; rather, we position it as a complementary approach that supports interactive inspection of multi-level structure, sample coverage, secondary metrics, and question-level evidence. In this context, our goal is not only to support precise lookup of individual metric values, but also to help users interpret performance patterns in relation to the underlying parent-child structure of knowledge domains. Rather than relying only on aggregate or tabular summaries, LLMVisExplorer offers a multifaceted performance analysis by exploiting the hierarchical knowledge domains underlying Q&A benchmark datasets. LLMVisExplorer's design is grounded in core requirements derived from both the NLP and visualization literature. First, it supports multifaceted, hierarchical performance decomposition, allowing users to assess LLM strengths and weaknesses across different levels of domain knowledge. Second, it provides an extensible metric encoding, visualizing not only accuracy, but optionally also secondary quantities, such as for instance response time, question difficulty, and question distribution through glyphs and dot patterns. Third, it enables interactive exploration in line with the "overview first, zoom and filter, details on demand" paradigm [42] -allowing users to drill down from global trends to individual benchmark questions.

With these design requirements in mind, LLMVisExplorer is built for clarity and scalability, ensuring interpretable visualizations even for large and complex datasets as well as multiple LLMs. For Q&A datasets, that do not directly expose a knowledge hierarchy, we further propose a semi-automated knowledge hierarchy generation approach, which can be refined by users to match domain-specific ontologies. Furthermore, LLMVisExplorer's plugin-based architecture allows users to extend the system with additional metrics or visual annotations as needed. We further evaluate LLMVisExplorer's utility through a controlled experiment as well as expert interviews in both the medical and NLP domains. Thus, within this paper, we make the following contributions:

- We introduce LLMVisExplorer, a system for multifaceted evaluation of LLMs, enabling drill-down analysis across hierarchical knowledge subdomains and incorporating multiple performance metrics such as accuracy, response time, and difficulty - within an extensible visualization framework.
- We propose a semi-automated approach to derive knowledge hierarchies for Q&A datasets that lack explicit structure, allowing user refinement to match domain-specific ontologies.
- We report initial user feedback based on a controlled experiment and expert interviews with domain professionals from both NLP and medicine.

We will further release all source code and data to foster transparency, reproducibility, and community adoption.

2. Related Work

We relate our work to developments in two technical-research subdomains. In the following section, we review works that evaluate and benchmark accuracy of natural language processing, followed by a section where we review knowledge visualization, particular in hierarchical or graph context.

LLM visualization. Recent advances in visualization research have led to the development of interactive systems that help interpret the performance of LLMs. These systems typically aim to overcome the limitations of standard benchmarking practices, which often reduce model behavior to a single aggregate score and provide limited visual support. For instance, the *LLM Comparator* [22] introduces a comprehensive visual analytics environment for side-by-side comparison of LLM responses. Built around automatic side-by-side evaluation (AutoSxS), it provides coordinated views for score distributions, rationale clusters, and customizable n-gram or regular-expression analyses. The system emphasizes bridging aggregated metrics with example-level inspection to support hypothesis formation and testing - a critical workflow for practitioners iteratively tuning and comparing models. While LLM Comparator excels at fine-grained, prompt-level analysis of model differences using automatic side-by-side evaluation, LLMVisExplorer is designed to provide a holistic and structured overview of a single model's strengths and weaknesses across hierarchically organized knowledge domains. By combining multi-metric aggregation with scalable, domain-aware visualization metaphors, LLMVisExplorer enables users to identify performance patterns not just in individual responses, but across entire subdomains - an analytical perspective that is not supported by existing prompt-centric tools.

Other recent tools, such as *EvalVerse* and *LMEvalVis*, propose dashboards that allow slicing evaluation results by task, domain, or capability, with support for comparative analysis. These tools often employ bar charts, heatmaps, or confusion matrices to communicate metric differences, but typically lack hierarchical encoding or explicit structural grounding in knowledge taxonomies. *LMdiff* [46] and *ChainForge* [2] offer prompt-oriented views that enable users to test model behavior across prompt variants and criteria, though they focus more on qualitative comparison than structured performance summarization. Spinner et al. presented *iNNspector*, facilitating systematic debugging through visual exploration across abstraction levels [44], and also introduced *generAItor*, a tree-in-the-loop system enhancing explainability during text generation [45]. Meanwhile, leading benchmarks and leaderboards such as HELM [26], Chatbot Arena [62], and OpenCompass provide only rudimentary visualizations - often limited to simple bar or radial plots - which hinders nuanced understanding of LLM capabilities across domains or tasks. Another promising research direction addresses reliability assessment in generated texts: *RELIC* utilizes self-consistency to highlight semantic inaccuracies and hallucinations [8], while Sevastjanova et al. proposed visual analytics to uncover biases in generated text sequences [41]. Together, these systems underscore a growing interest in visualization as a means for transparent and multifaceted LLM evaluation. However, most lack structured representations of domain knowledge hierarchies combined with interactive metaphors for drilling into performance variations on complex Q&A benchmarks.

LLM evaluation is also important for model deployment. Mitchel et al. call for new technology to better communicate risks of deployed language models [29]. Their proposed model cards, should be released together with a model, and contain details regarding the model's benchmarked evaluation characteristics. Similarly to as we see LLMVisExplorer, Mitchell et al. see model cards as a step towards the responsible democratization of ML models, by increasing transparency into how well models perform. Raj et al. even demand specific AI audit trails, which also contain model cards, but are meant as an internal audit framework to assess risks of AI models before their deployment [34]. While some researchers explicitly underline the importance of visual representations in model cards [13], LLM evaluations rarely make use of visualizations beyond simple bar charts or spider charts, e.g., [4, 51, 37, 50, 33, 27]. In their recent survey, Chang et al. summarize the current state of LLM evaluation, which, however, lacks methods accessible to a non-expert user base, beyond the discussed coarse depictions of accuracies. Based on these observations, we see a clear need for more visualization in the context of LLM evaluation.

Q&A benchmarks. Evaluating the capabilities of LLMs through Q&A datasets has become standard practice in natural language processing. These datasets vary in domain coverage, size, and question complexity, often aligning with hierarchical knowledge structures that facilitate nuanced model assessment.

To assess comprehensive language understanding, benchmarks such as GLUE and HELM have been introduced. GLUE (General Language Understanding Evaluation) is a collection of nine benchmarks designed to evaluate general language understanding [54]. HELM (Holistic Evaluation of Language Models) evaluates LLMs across diverse scenarios, including question answering, summarization, and sentiment analysis, whereby questions are structured within 42 subdomains [26].

For general knowledge and reasoning, datasets like HotpotQA, WikiHop, TriviaQA, and ComplexWebQuestions have been developed. HotpotQA offers a diverse set of 113,000 questions derived from Wikipedia articles across various subjects [58]. WikiHop is another Q&A dataset based on Wikipedia, containing approximately 51,000 questions [56]. TriviaQA spans a broad range of subdomains, including history, literature, and science [21]. ComplexWebQuestions sources its 34,689 questions from web queries, reflecting real-world information needs across various domains [48].

In the biomedical domain, BioASQ focuses on various subdomains such as genomics, diagnostics, and treatment. As of 2020, it comprises approximately 3,243 training questions and at least 500 test questions [52]. PubMedQA is another dataset designed for biomedical research question answering, which comprises 1,000 expert-annotated questions sourced form PubMed [19]. To evaluate language models across a wider range of knowledge domains, the Massive Multitask Language Understanding (MMLU) benchmark was introduced, consisting of approximately 16,000 multiple-choice questions spanning 57 subdomains, including mathematics, philosophy, law, and medicine [17]. Building upon this, MMLU-Pro offers a more robust and challenging evaluation

by integrating complex, reasoning-focused questions from 14 diverse domains [55].

For advanced and contamination-free benchmarks, GPQA and LiveBench have been developed. GPQA (Graduate-Level Google-Proof Q&A) targets advanced topics in biology, physics, and chemistry, consisting of 448 questions designed to be challenging even with unrestricted web access [35]. LiveBench is designed to mitigate test set contamination by releasing new questions monthly, sourced from recently released datasets, arXiv papers, news articles, and IMDb movie synopses. Each question has verifiable, objective ground-truth answers, allowing for accurate and automatic scoring without relying on LLM judges [57].

Many of these datasets inherently provide subdomain labels or are structured to facilitate the construction of hierarchical knowledge representations. For instance, BioASQ's categorization into specific biomedical subdomains and GPQA's focus on distinct scientific domains allow for natural mapping to hierarchical knowledge structures.

Hierarchical and instance-level visual analysis. LLMVisExplorer is also related to the broader literature on hierarchical visualization and text-oriented visual analytics. Established hierarchy visualizations, including node-link diagrams, treemaps, icicle plots, sunburst charts, and matrix-style representations, offer different trade-offs among compactness, label readability, structural explicitness, and quantitative comparability [38]. Space-filling techniques such as treemaps and icicle plots [20, 24, 39] are particularly relevant because they can represent large hierarchies compactly, but they may also make labels, local adjacency, or precise metric comparisons difficult in dense views. Text and instance-level visual analytics systems further motivate coordinated workflows that connect aggregate patterns to individual examples. LLMVisExplorer builds on these principles by combining a hierarchy-aware overview with embedded quantitative encodings and drill-down access to individual benchmark questions. Our design should therefore be understood as one point in a broader design space rather than as a universally optimal hierarchy visualization.

3. System Design Requirements

Within this section, we briefly discuss the system design requirements, which underlie our visualization design introduced in Section 4. These requirements are based on both, demands expressed in the NLP literature, as well as good practices described in the visualization literature.

DR1: Multifaceted Knowledge Assessment. Since the knowledge capabilities of LLMs are progressing fast, and to some extent towards those of a human [33], we take design inspiration from the current state of the art in knowledge assessment as done in education. Knowledge assessment is a process for determining the nature and extent of human learning and development [30]. In this context, Q&A benchmarks are directly related to summative knowledge assessment, which is concerned with measuring learning outcomes. Summative knowledge assessment is widely considered to be effective when targeted towards individual courses of a curriculum [47]. Within these, knowledge is typically assessed for different subdomains, such as for instances courses in a university degree program. Similar to humans, LLMs might be doing well in some subdomains, while performing subpar in others. This is also the main reason, why Chang et al. call for a multifaceted evaluation of LLMs [7], which is not adequately met by merely combining multiple Q&A datasets, as is usual practice. Thus, for comprehensive understanding of LLM performance, users need to understand the performance for different subdomains, whether the subdomains are well populated with questions, and which questions in particular the LLM did (or did not) accurately answer.

DR2: Extendable Metric-Based Outcome. For effective assessment in LLMs, we should further focus on important learning goals and consider what is considered evidence for human learning. From the main groups of evidence for human learning, such as written work, performance, and presentation [30], we see performance as measured based on correct test responses, as the most obvious direct evidence for LLM *learning*. While written work could certainly also be considered, it is more difficult to rate, as it mixes text generation and knowledge capabilities [28]. Therefore, we, and many other researchers, focus on evaluating factuality by means of Q&A data sets [4, 25, 33, 31, 5]. In addition to the main accuracy, additional metrics must be integrable, such as F1 or precision and recall, to support a more nuanced analysis of model performance on structured-answer or multi-label tasks. Moreover, recent work has pointed out the limitations of accuracy in capturing complex failure modes of LLMs [36, 61]. Metrics like hallucination rate [18], calibration error [15], response latency, and difficulty-adjusted correctness provide complementary perspectives on model behavior that go beyond binary correctness, and shall therefore be easily realizable.

DR3: Comparative Analysis. While the above observations have mainly been made by focusing on the evaluation of a single LLM, it is customary to compare a new LLM's performance against several other LLMs - possibly through leader-boards. One of the most recent endeavors in this area is LiveBench, a hierarchical Q&A dataset, that is to be regularly updated in order to reduce the risk of training data contamination [57]. Based on such and similar leader-boards, we see, besides the evaluation of a single LLM on one or multiple Q&A datasets, the comparative analysis of multiple LLMs on one or multiple Q&A datasets, as an important system design requirement. In the single LLM usage scenario, LLMVisExplorer shall uncover an LLM's limitations, which can give model developers useful clues on where improvements are necessary, which could in the simplest case just be selectively increasing the amount of training data. In the multiple LLM usage scenario, we see LLMVisExplorer's role more targeted towards AI users and policymakers, which require transparency to decide for or against a single model.

DR4: Interactive Analysis. The diversity and size of modern Q&A benchmark data sets, does not only pose a limit when considering single number accuracy evaluation. Any type of visualization will be challenged by the data complexity to be visualized. Therefore, several authors have called for bringing in interaction when evaluating LLMs [22, 9]. We follow that call and target interactive exploration in line with the "overview first, zoom and filter, details on demand" paradigm [42]. Thus, users

can move from global trends to individual benchmark questions, and thus gain more knowledge and trust about LLMs knowledge capabilities. Beyond exploratory interaction metaphors, we also target the creation of test sets. LLMs are nowadays a component of software systems and as such they need to be tested and evaluated in the software development cycle. LLMVisExplorer can support by creating individual test sets, or by hierarchically structuring Q&A data sets provided without hierarchies. Thus, LLMs can be evaluated for desired application cases, and it also becomes easier to ensure that the questions are not part of their training set.

4. Visualization Design

To address the above stated design requirements DR1-DR4, we propose LLMVisExplorer. LLMVisExplorer's backbone is a hierarchical visualization inspired by mind maps (DR1). However, in contrast to frequently used mind maps, it arranges knowledge hierarchies in a more compact, horizontal layout - departing from the typically radial and sparse design - while embedding quantitative performance metrics (DR2) to support a comparative (DR3) and interactive analysis (DR4). In the following we discuss the visualization design in detail and relate it to our system design requirements.

4.1. Visualization Layout

To support multifaceted knowledge assessment (DR1) of different knowledge subdomains, our hierarchical visualization layout is informed by the knowledge hierarchy of the underlying Q&A datasets, with the containing questions assigned to the leaf nodes of this hierarchy. The layout draws on the explicit parent-child organization commonly used in mind maps, which have been applied in problem-based learning, small-group teaching, and examination tools [16]. Mind map layouts are known to engage learners [10, 40] and to positively affect learning outcomes in different domains [3, 14].

Visualization metaphor. Traditional approaches for visualizing hierarchies often rely on node-link diagrams, where items are shown as nodes connected by lines that represent parent-child relationships. While intuitive for small trees, these layouts struggle to scale, as links and nodes quickly compete for limited space, leading to cluttered and inefficient visualizations as hierarchies grow deeper and broader [38]. To overcome these limitations, space-filling visualizations such as treemaps [20] and dense icicle plots [24] have been introduced. These techniques allocate visual space proportionally to hierarchical elements, allowing for more compact and scalable displays. In designing LLMVisExplorers, we drew inspiration from the classical icicle plots introduced by Kruskal and Landwehr [24], which offers several compelling advantages over alternative layouts.

While treemaps are highly space-efficient [39], they convey hierarchies only implicitly, and do often suffer from readability issues: boxes can become too small to display meaningful labels, especially in deeper hierarchies [20]. Similarly, stacked bar charts lack the expressive power to convey hierarchical depth, making them less suitable for multifaceted analysis [12]. Radial charts on the other hand, though visually striking and sometime used for LLM visualization [60], distort spatial relationships and hinder accurate comparisons due to their circular geometry [6]. Sunburst charts are useful for hierarchical exploration, but the orientation of the labels (rotating around the center) makes them difficult to read. In addition, they do not easily allow for allocating more space if the hierarchy requires it [23]. In contrast, our dense, horizontal hierarchy visualization is designed to preserve the logical structure of hierarchies while maintaining consistent orientation and scale across hierarchy levels. This facilitates allocating more vertical space if needed. Furthermore, labels can remain predominantly horizontal, avoiding the rotated label orientations common in radial alternatives such as sunburst charts.

Layout computation. To visualize Q&A datasets hierarchically, we represent the data as a rooted tree $T = (V, E)$, where each node $v \in V$ has an associated depth $d(v) \in \mathbb{N}_0$, and the maximum tree depth is denoted by $D = \max_{v \in V} d(v)$. Each node v is assigned a bounding box:

$$B(v) = (x_v, y_v, w_v, h_v)$$

where x_v and y_v define the top-left position, and w_v, h_v define the width and height. Furthermore, to guide both label placement and node connections, each node computes an aggregate bounding box over its children:

$$B_C(v) = \mathrm{AABB}\left(\bigcup_{c \in C(v)} B(c) \right)$$

This axis-aligned bounding box enables smooth spatial transitions and ensures consistent subdivision geometry across hierarchy levels.

When designing our knowledge hierarchy visualization, we have experienced figure height, as opposed to figure width, as the main constraining factor for all handled Q&A datasets. Therefore, we have decided to bisect the knowledge hierarchy at the root node into halves and visualize one half oriented towards the left and one towards the right side. This design decision is in line with Ondov et al.'s findings, which report cases with significant task performance improvements when arranging small multiples in a mirror-symmetric fashion [32]. Thus, the horizontal coordinate x_v of a node is computed based on the depth of the node:

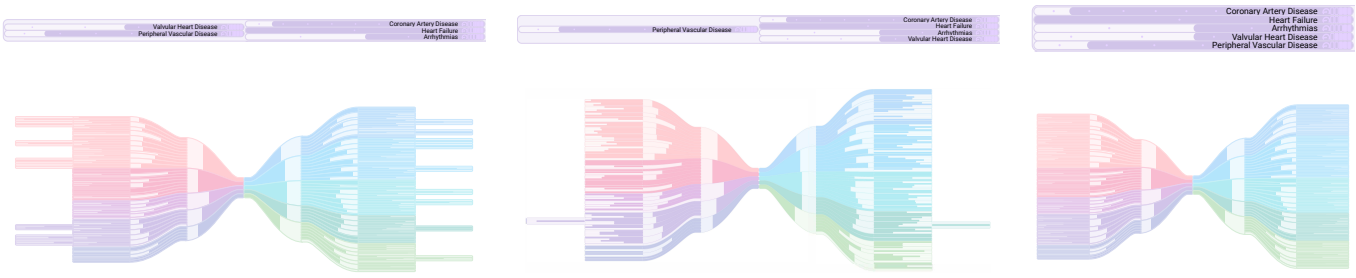
$$x_v = \pm\, d(v) \cdot \ell$$

where ℓ is the fixed horizontal layer width, and the sign distinguishes between the left and right branches of the bisected layout.

Vertical placement of nodes within each layer is based on reverse level-order traversal. A linear interpolation function is used to vary vertical spacing by depth:

$$s(d) = s_{\min} + (s_{\max} - s_{\min}) \left(1 - \frac{d}{D}\right)$$

Here, $s(d)$ defines the vertical gap at depth d, with $s_{\min}$ and $s_{\max}$ denoting the minimal and maximal spacing, and D denotes the maximum depth of the hierarchy, that is, the greatest depth value among all nodes in the tree.



(a) $k = 3$ (b) $k = 4$ (c) $k = 5$

Fig. 2: Comparison of different values for leaf stacking parameter k. Larger values for k result in more leaf nodes stacked (*top row*), which trades overall figure width for height (*bottom row*). Parent-child relations on leaf level are emphasized by parents encapsulating child nodes.

Node height and font size also interpolate linearly with depth to maintain visual hierarchy and space economy.

Relationship representation. To achieve a compact and continuous visual flow, edges are not drawn explicitly. Instead, the relationships between a parent node v and its children $C(v)$ are rendered as Bézier-shaped connector regions. Each side of such a region is defined by a cubic Bézier curve:

$$\mathbf{B}(t) = \sum_{i=0}^{3} \binom{3}{i} (1-t)^{3-i} t^i \, \mathbf{P}_i, \quad t \in [0,1]. \tag{1}$$

The control points $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3 \in \mathbb{R}^2$ are derived from the axis-aligned bounding box $B(v) = (x_v, y_v, w_v, h_v)$ of the parent node and the aggregate bounding box $B_C(v) = (x_c, y_c, w_c, h_c)$ of its children. Specifically:

- $\mathbf{P}_0 = (x_v, y_v)$ is the top edge of $B(v)$,
- $\mathbf{P}_1 = (x_v + (1-\sigma)\Delta x, y_v)$ offsets horizontally toward C,
- $\mathbf{P}_2 = (x_c - \sigma\Delta x, y_c)$ bends into C,
- $\mathbf{P}_3 = (x_c, y_c)$ is the entry point to $B_C(v)$,

where $\Delta x = x_c - x_v$ is the horizontal distance between the layers, and $\sigma \in [0,1]$ is a smoothness parameter controlling the curvature. The same principle applies to the bottom side of the connector, forming a closed region between the top and bottom Bézier paths. This continuous envelope reinforces the hierarchical structure while maintaining visual clarity in dense layouts. The resulting horizontal "bow-tie" layout preserves the parent-child structure while creating a central region that can accommodate complementary information without further increasing the vertical extent of the hierarchy. In our examples, this space is used for summaries such as Bloom's taxonomy and overall model results alongside the hierarchical performance view (see Fig. 1). This design involves a trade-off: the straight-bar BarVis alternative supports more precise isolated magnitude judgments, as reflected in our controlled experiment, whereas the curved connectors provide a continuous visual link between hierarchy levels and allow hierarchy, performance, and complementary information to be integrated within a compact layout. We therefore do not claim the curved layout to be optimal for precise quantitative comparison; it is intended for overview-oriented hierarchical exploration in which structural context and quantitative values are inspected together. **Leaf node representation.** While our bisected layout reduces figure height roughly by 50%, height remains the main limiting factor for both screen-based and print-based use. We therefore allow leaf nodes to be horizontally stacked beneath their parent in order to better match typical aspect ratios and further reduce overall height. Importantly, this stacking is restricted to the leaf level, that is, to nodes that already share the same depth in the hierarchy. Thus, the layout does not alter the structural ordering of internal nodes, but only compacts the presentation of terminal subdomains. To preserve parent-child readability despite this compaction, stacked leaves remain spatially grouped and visually encapsulated by their parent node. This compact arrangement therefore trades some strict depth regularity at the leaf level for improved space efficiency. Given n leaves and a maximum number of horizontal stacks k, we arrange leaf nodes in a 2D grid beneath their parent as follows:

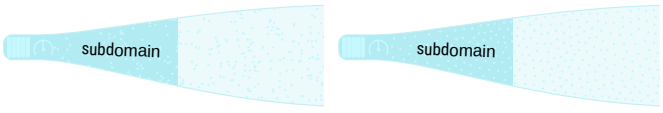
$$c = \left\lceil \frac{n}{k} \right\rceil$$

as the number of columns. The position of each leaf node v is computed as:

$$x_v = x_p + j \cdot (\ell + \delta_x), \quad y_v = y_p + i \cdot (h_v + s(d))$$

where (i, j) is the row/column index of v, x_p, y_p are the parent node's coordinates, and δ_x is the inter-column spacing. Fig. 2 illustrates this concept for $k = 3, 4, 5$. The stacked leaf nodes are encapsulated by their parent nodes to reinforce parent-child relations despite the compact arrangement. In our experiments, a conservative default of k=3 provided a good balance between reduced figure height and readable grouping across the handled Q&A datasets, while still allowing the parameter to be adjusted for different hierarchy sizes and display settings.

Another leaf-related consideration, which affects figure size, is whether to show empty leaf nodes, that is, those subdomains for which the Q&A dataset does not contain any questions. The relative amount of empty leaf nodes can become large, especially for small sample sizes in a deep knowledge hierarchy. Empty leaf nodes play an important role, when analyzing Q&A datasets, as they indicate subdomains which are not represented by the dataset. However, when the main goal is to see how a LLM



(a) Number of questions per subdomain represented as random noise pattern (b) Number of questions per subdomain represented as blue noise pattern

Fig. 3: Comparison of randomly *(a)* vs. blue noise distributed points *(b)* for encoding the number of questions per subdomain. Due to the blue noise criterion, no point aggregations influence the perceived density.

performs on a given benchmark dataset, we consider them less relevant. Therefore, we have decided to omit empty leaf nodes per default from our visualization, while still giving users the opportunity to override this default on demand.

4.2. *Quantity Encoding*

To enable the assessment of metric-based outcome in an extendable way (DR2), and to be able to compare these quantities across subdomains and LLMs (DR3), we have integrated the visualization of quantitative information into the above layout.

Primary quantity. As we consider LLM performance per subdomain to be the primary quantity, it is the most important information communicated by LLMVisExplorer. We therefore visualize accuracy per subdomain through horizontal bars embedded into the curved subdomain regions. Rather than displaying these values as detached straight bars, we intentionally integrate them into the hierarchical flow of the layout. This choice preserves structural continuity between parent and child nodes and supports an integrated overview of the knowledge landscape within a compact visualization. At the same time, this design introduces a perceptual trade-off: compared with a conventional straight bar representation, curved regions may reduce the precision of isolated magnitude judgments. We chose this encoding because our goal is not only precise value lookup, but the joint communication of hierarchical structure and performance variation across subdomains. We note that this integrated structural overview is not itself the end goal of the system; rather, we hypothesize it supports higher-level analytical tasks such as locating weak subdomains or comparing models. Our evaluation provides perceptual and preference-based evidence for this design choice, but does not measure whether it improves objective performance on such realistic analytical tasks.We therefore do not claim such a performance benefit based on the current results. We would like to emphasize that this is an aggregated measure, computed for all subdomains in a depth-first traversal based on the correctness of the individual questions. While this enables direct performance comparison within each half of the knowledge tree, comparison across halves is less direct. Furthermore, a bar does not provide exact quantitative values without appropriate axis labels. We therefore additionally display the accuracy numerically. As we regard this number as a supporting aid rather than the primary encoding, we traded some contrast for the overall visualization design and computed a perceptually harmonious yet high-contrast color (see Sec. 4.3 for details).

Secondary quantities. To visualize secondary, less important, quantities, we have integrated an extendable icon layout mechanism. These icons are intended as optional supplementary cues rather than primary readout channels, and their legibility depends on sufficient display space and contrast. This mechanism allows for visualizing icons near the head of the respective node, whereby the icon representation encodes the desired quantity - also aggregated per subdomain in a depth first traversal. To place these icons, but also the text labels associated with the nodes, along our hierarchical structure, we compute a centerline between each parent node and its children. This centerline is also defined as a cubic Bézier curve using control points interpolated between the vertical midpoints y_v and y_c of the respective bounding boxes, with horizontal control point offset proportional to the layer distance $\Delta x = x_c - x_v$. The curve is given by Equation 1, where $\mathbf{P}_0 = (x_v + r, y_v)$, $\mathbf{P}_1 = (x_v + (1-\sigma)\Delta x, y_v)$, $\mathbf{P}_2 = (x_c - \sigma\Delta x, y_c)$, and $\mathbf{P}_3 = (x_c, y_c)$. The parameter r accounts for rounded edge radius, and $\sigma \in [0, 1]$ controls the curvature smoothness. The thus computed centerline serves as a guide to align all icons for secondary quantities, whereby their order shall be decided on importance.

To showcase the extendable icon layout mechanism within LLMVisExplorer, we have exemplarily integrated three secondary quantities, which can be optionally displayed (see Fig. 4). These are, the difficulty rating of the questions assigned to the node in question, as well as the average response time of the LLM. Furthermore, we have integrated a simple hallucination rate, which is defined as the percentage of generated outputs that contain factual errors or unsupported claims:

$$\text{Hallucination Rate} = \frac{\text{Number of hallucinated responses}}{\text{Total number of evaluated responses}} \times 100$$

To depict question difficulty, ranked on a 5-point Likert scale, we opted for a discrete progress bar. To depict response time, we instead suggest using a speed gauge, which shows a value normalized over all displayed questions and and at the same time aims to support icon recognizability under moderate visual density. The hallucination rate is depicted by a bull's eye icon, whose degree of filling communicates this quantity. Again, we only see these quantities as examples to showcase our layout mechanism, and encourage LLMVisExplorer users to integrate their quantities and icons. The effectiveness of these icon encodings depends on available node size, contrast, and visual density. In dense static views, especially at the leaf level or in multi-model color settings, icon visibility can decrease substantially. Accordingly, we treat them primarily as optional

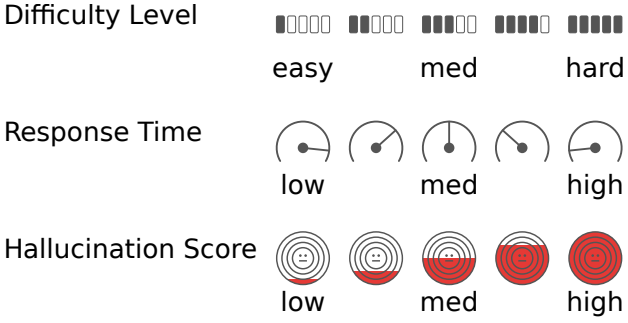


Fig. 4: Icons depict optional secondary quantities, such as difficulty level, response time, and hallucination score.

supplementary annotations for obtaining a quick overview of secondary quantities, rather than as encodings intended for precise quantitative comparison. Users can selectively enable or disable these encodings depending on the analysis task, allowing the view to be simplified when secondary metrics are not needed. In dense static overview figures, these icons may remain difficult to distinguish because substantially enlarging them would compete with node labels and the hierarchical structure. In the interactive system, users can instead pan and zoom into regions of interest and access the underlying secondary metric values through details-on-demand tooltips. For static presentation, the icons may be omitted when space is insufficient. Reference examples of the three icon encodings are provided in Fig. 4.

Comparative visualization. While Fig. 1 supports the multi-faceted evaluation of a single LLM, it does not allow for performance comparisons of multiple LLMs based on one or several Q&A datasets (DR3). To also support this use case scenario, we have extended LLMVisExplorer by being able to communicate the performance of multiple models. To do so, we horizontally split the performance bar into n bars, where n is the number of visualized LLMs. Furthermore, while for the single-model use case scenario, we have chosen hue, as the most discriminative color feature, to visually encode subdomain, we have decided to use hue to visually encode model in our multi-model scenario (see Sec. 4.3 for details). Fig. 8 shows an example of the resulting visualization. As can be seen, it now becomes possible to compare models per subdomain, while still being able to compare their overall performance through the conventional bar chart used in most LLM performance comparisons.

Question distribution. The accuracies per subdomain on a given Q&A dataset are only fully interpretable, when considering the actual number of questions associated with a specific subdomain - as also considered in statistical power analysis. Therefore, we also have to visualize this quantity, whereby we consider it of subordinate importance rather than the performance score. Accordingly, our goal is not precise numeric lookup of sample counts within the overview, but a compact visual indication of relative coverage, sparsity, and imbalance across subdomains. Since the number of questions per subdomain can be interpreted as a density value, we decided to choose commonly used density visualizations to display it. As point-based density plots are both widely used, and non-conflicting with the used bar chart, we decided to visualize the number of questions as a dot plot overlaid over a subdomain. This choice intentionally favors qualitative density assessment over exact counting, while avoiding interference with the primary bar-based performance encoding. Since we have both dimensions within a subdomain as free variables, we have decided to realize our dot plots as an extension to Blue Noise Plots [53]. By facilitating a blue noise distribution, we are able to evenly distribute the points within each subdomain, without resulting in unwanted clustering effects. While the original Blue Noise Plots operate within a fixed rectangular layout, we generalize the approach by supporting point distribution within arbitrary 2D shapes. To achieve this, we parse our curve-based node layout geometry, and construct polygonal regions from these, in which we uniformly sample candidate points, before applying blue noise optimization through Lloyd relaxation. This extension enables semantically meaningful and spatially constrained visual encodings that adapt to complex visual structures, as given by our subdomain regions. Fig. 3 shows a comparison of our blue noise distribution as compared to a unconstrained random distribution, where overlap and clustering affects the perception of question density.

A practical consideration of the proposed design is scalability. LLMVisExplorer is intended for hierarchies of moderate size, where users can still benefit from an integrated overview of structure and performance within a single view. For substantially larger domain hierarchies, a fully expanded static visualization may become visually dense and reduce the readability of labels, icons, and local structural relations. In such cases, LLMVisExplorer is intended to be used progressively through interaction, e.g., by collapsing or expanding subtrees, filtering relevant branches, omitting empty leaves, selectively showing secondary quantities, and relying on details-on-demand views. Thus, the system does not aim to display arbitrarily large hierarchies fully expanded at once, but to support scalable exploration through a combination of compact layout and interactive abstraction. In practice, the benchmark datasets used in our examples range from 210 to 4,176 questions, including SciQ and PubMedQA with 997 questions each. At this scale, LLMVisExplorer supports effective exploration through the combination of the compact overview and interactive mechanisms described above; however, the limiting factor for a fully expanded static view is primarily the size and depth of the hierarchy rather than the number of questions itself. In our examples, hierarchies with roughly up to 100 leaf nodes remain suitable for a single static overview, while larger or more fine-grained hierarchies benefit from progressive exploration and pruning.

4.3. Color Coding

As in all visualization scenarios, an appropriate color coding is mandatory to obtain an interpretable visualization. Therefore, we have developed a data-centric and a model-centric color coding strategy.

Data-centric colors. The data-centric color coding is meant for single-model evaluations. In this case, we have chosen a color scheme, which allows for differentiating subdomains based on the hues in Google's material color scheme. In each subdomain, we employ a brighter shade of that hue for the background C_b and, a darker shade C_d for the accuracy bar. To communicate accuracy number, secondary quantity of icons and question distribution, we need an additional color which has sufficient contrast to both, background and bar color - as at least the dots representing question distribution will cover the background and bar. At the same time, we want to match the hue of the respective subdomain, to make the subdomain association of the quantities clear. While this constraints the degree of contrast, we can achieve, we have taken this design consideration, as all quantities represented with this color are of secondary importance only. Thus, for each subdomain we compute an additional perceptually harmonious yet high-contrast color between C_b and C_d, we convert both to HLS space and average their hue and saturation. The lightness component is blended with its inverse to enhance contrast:

$$L = (1-\alpha)\cdot\frac{L_b+L_d}{2} + \alpha\cdot\left(1-\frac{L_b+L_d}{2}\right),$$

where $\alpha \in [0,1]$ controls the contrast strength. The resulting HLS tuple is then converted back to RGB to yield the final contrast color.

Model-centric colors. The model-centric color coding is meant to be used for multi-model evaluations, where it is important to clearly associate each bar with a specific model. Therefore, we use the hue, as the perceptually most dominant color parameter, to depict models rather than subdomains (see Fig. 8). To also allow for differentiating the latter, we use alternating shades from Google's material color scheme for their background color.

5. Interaction Design

The expert evaluations showed that participants could use the visual overview to identify high-level performance patterns, including subdomain-level strengths and weaknesses and differences in question coverage. Interaction then plays a central role in enabling deeper analysis of complex benchmark results. In particular, it allows users to navigate large hierarchies progressively, inspect exact values and question-level evidence, compare relevant subdomains, and construct focused test sets for downstream evaluation workflows. In the following, we describe the main interaction mechanisms supported by LLMVisExplorer.

5.1. Result Exploration

The interaction design of LLMVisExplorer follows Shneiderman's visual information-seeking mantra [42] to allow for interactive exploration, and thus fulfill design requirement DR4. Upon loading, users are presented with the high-level *overview* of model performance across knowledge domains in our horizontal hierarchy visualization (see 1 in Fig. 1). This overview enables quick orientation by visualizing overall accuracies, response times, difficulty levels, and hallucination metrics directly within the subdomains. Users can then *zoom and filter* into specific subtrees via intuitive panning and zooming interactions over the visualization layout (see 2 in Fig. 1). A dedicated "Reset Zoom" button allows users to quickly re-center the view. Clicking any hierarchical region (node) dynamically filters the view to reveal more detailed data: it highlights the selected subdomain and displays the associated questions together with expected and given answers in the right question pane (see 3 in Fig. 1). For *details-on-demand*, an information-rich tooltip system provides full per-node metadata on hover, including accuracy broken down by difficulty level, average response time, and hallucination rate. Together, these interactions support seamless exploration across levels of abstraction - from global performance patterns to individual question-level insights - while maintaining responsiveness and interpretability.

5.2. Hierarchy Generation

Many Q&A datasets already contain some form of topic labeling (e.g., MMLU-Pro, GPQA, or BioASQ), which can be directly used within LLMVisExplorer. However, in some cases

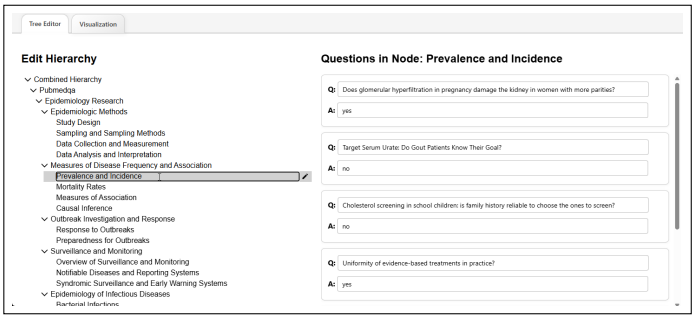


Fig. 5: The *Tree Editor* enables users to create or change knowledge hierarchies, and associate new or existing questions with subdomains.

the provided hierarchy labels are either too coarse (e.g., single-level domains) or too fine-grained and inconsistent to directly support a meaningful hierarchical visualization. To this end, LLMVisExplorer allows users to inspect and revise the underlying knowledge hierarchy within its *Tree Editor* interactively (see Fig. 5). Within the editor, users can restructure the tree (adding, renaming, or removing nodes) and reassign individual questions to different subdomains via drag-and-drop. Questions that remain unassigned, shall be reassigned or are newly added, can be manually reassigned using a dedicated interface - ensuring to be able to exploit manual and automated question generation [43, 57] to have full control over the final test set and the structure of our visualization.

Because the knowledge hierarchy is stored independently of any individual model's results, as a mapping from questions to subdomains, the same edited hierarchy is applied identically when aggregating results for every model shown in a given visualization; an edit therefore cannot introduce inconsistent groupings across the models being compared. All displayed quantities, including accuracy, question density, and secondary icon metrics, are aggregated per subdomain through a depth-first traversal (Sec. 4.2) and are recomputed for the affected node and all of its ancestors after an edit. In the current implementation, this recomputation is performed by regenerating the visualization from the edited hierarchy through our offline rendering pipeline, rather than live within the browser-based editor. Wiring the editor's confirmation step directly to in-browser recomputation, together with the ability to export and reload an edited hierarchy for later reuse, is an active engineering effort that we identify as near-term future work.

For datasets that lack an explicit hierarchy, we provide an LLM-assisted hierarchy generation pipeline. By prompting a state-of-the-art language model, users can derive top-down topic trees and textbook-style outlines, which are parsed into multi-level knowledge hierarchies. These hierarchies are fully editable, and expert oversight remains encouraged, but the automation significantly lowers the barrier for creating new stratified benchmarks. This functionality is not only key for interpreting existing datasets, but also for curating custom test sets that reflect targeted knowledge areas, thereby supporting iterative refinement and reproducible evaluation workflows.

Table 1 reports node counts, leaf counts, and depth for the hierarchies underlying the datasets used throughout this

Table 1: Hierarchy statistics for the processed Q&A datasets used in our visualizations. *Existing* taxonomies are supplied by the dataset, whereas *generated* hierarchies are produced or extended by our LLM-assisted pipeline. Node and leaf counts refer to the processed hierarchy used by LLMVisExplorer; question counts indicate the questions retained for visualization. Depth is measured from the root (D=0).

Dataset	Taxonomy origin	Nodes	Leaves	Depth (D)	Questions used
US Bar Exam (subjects only)	Existing	7	7	1	622
US Bar Exam (+ generated subtopics)	Existing + generated	75	68	2	622
MMLU subset	Existing + generated	51	37	2	210
SciQ (pruned, as displayed)	Generated	129	93	3	997
SciQ (full, before pruning)	Generated	218	160	3	997
PubMedQA	Generated	511	400	3	997
MedMCQA	Generated	643	490	3	4,176

paper, distinguishing datasets with an existing top-level taxonomy (for example, the seven official Multistate Bar Exam subjects) from hierarchies generated or extended through the LLM-assisted pipeline described above. Shallow or moderately sized hierarchies, such as the 7-subject Bar Exam breakdown and the processed MMLU subset used in our examples, fit comfortably within a single fully expanded overview. Generated hierarchies that further subdivide subtopics, such as PubMedQA (400 leaves) or MedMCQA (490 leaves), exceed what can be legibly shown fully expanded at depth 3. For these cases, LLMVisExplorer relies on the pruning of empty leaves described in Sec. 4.2 - which alone reduces SciQ from 218 to 129 nodes - together with the interactive collapsing, filtering, and details-on-demand mechanisms described in Sec. 5.1 to keep the displayed view legible. Based on the datasets in Table 1, we consider hierarchies of up to roughly 100 leaf nodes displayable as a single static overview, while larger generated hierarchies benefit from progressive, interaction-driven exploration rather than a single fully expanded view.

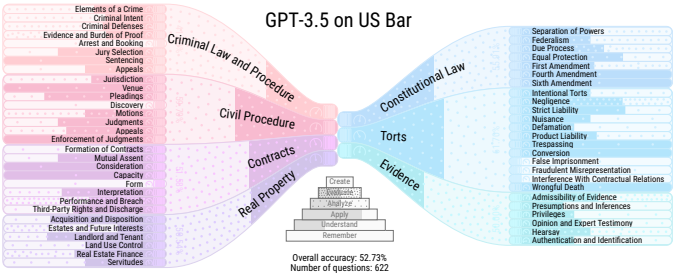


Fig. 6: LLMVisExplorer for GPT-3.5's results on the US Multistate Bar Examination questions grouped by subdomain.

6. Bloom Taxonomy

Besides the main visualization layout, and the two essential quantities, that is, accuracy and number of questions, we believe that LLMVisExplorers are also useful to visualize further information relevant for the evaluation of LLMs. In this section, we would like to entertain this idea and outline three such examples, to inspire the development of future extensions to be integrated into LLMVisExplorers.

According to Tamkin et al., one central question regarding LLMs, is whether they exhibit *understanding* [49]. Therefore, to assess the quality of LLMs, we should consider the dimensions of learning, which range from conceptual understanding, over creativity to problem-solving [47]. As, these dimensions are also highly relevant for human learners, in the past, several taxonomies for classifying learning content, have been proposed. Bloom's Taxonomy is perhaps the most widely known and used classification system for knowledge questions [1]. It is a hierarchical framework that classifies educational learning objectives into six different levels of complexity and specificity: Remembering, Understanding, Applying, Analyzing, Evaluating, and Creating. It has been used extensively in education, from K-12 classrooms to higher education and professional development. While several other such taxonomies exist (e.g., Structure of Observed Learning Outcomes (SOLO) Taxonomy, Webb's Depth of Knowledge, Fink's Taxonomy of Significant Learning, and Marzano's Taxonomy), due to its widespread use, we have decided to choose Bloom's Taxonomy to classify the type of knowledge gain for each question, contained in the processed Q&A datasets.

For the semantic initialization of this hierarchy, we use LLM-assisted classification prompts to assign question-level metadata, in particular cognitive categories based on Bloom's taxonomy and difficulty ratings. In our implementation, each question is processed independently via OpenAI's API using dedicated prompts for multiple-choice and free-form questions, with a fresh session per question to reduce contextual carry-over effects. The exact prompt templates used for difficulty estimation and Bloom-category assignment are provided in the supplementary material (Sec. 2). We use these labels as a structured basis for grouping and organizing questions into an initial hierarchy, which can then be interactively revised by users when domain-specific adjustments are needed.

Thus, we consider the dimensions of learning by classifying all questions according to Bloom with an LLM, and stratify our evaluation also based on Bloom's levels. To visualize Bloom's taxonomy in the context of the evaluated LLM, we display the accuracy for each level in a pyramidal fashion, as it is usually chosen when displaying Bloom's taxonomy (see Fig. 1 (bottom)). Naturally, to judge the reliability of this classification, it is again important to estimate how many questions are associated with each Bloom level. To do so, we employ the same blue noise visualization as we have used to display the number of questions per subdomain, while each point accounts for 1% of all samples

to facilitate the readability.

7. Implementation Details

Within this section, we briefly describe how we have implemented LLMVisExplorer. Due to the widespread use of Python in the ML community, we have also used it to implement LLMVisExplorer. We will first describe, how Q&A datasets can be imported and stored, before we describe how we obtain results for these from the LLMs. Finally, we will provide some details on implementing the visualization. All generated data and source code, for processing data and generating LLMVisExplorer, will be available on GitHub in case of acceptance.

7.1. Q&A Processing

To leverage automatic processing of different Q&A datasets in the visualization pipeline, they first need to be transformed into a common data structure. To define what information fields are necessary, we combined the knowledge obtained from probing numerous Q&A datasets, whereby we for instance considered the HELM dataset [26] and the datasets provided by the University of Freiburg's Algorithms and Data Structures Group[1]. Based on these reviewed datasets, our data structure contains fields for the question text, short and long versions of the answer, context information, the question type, fields for Bloom classification and difficulty rating, the topic of the question, and if applicable, the answer options. Depending on the raw unprocessed dataset, not every field must necessarily be filled. As such, the type of content for short and long answers, as well as for the topic and context, differs depending on question type and available information. The short answer may be an index in the case of multiple choice questions or a single word or short statement where questions are expected to be answered in such a way. The long answer consequently can be the text of the correct choice in multiple choice datasets, a short answer with reasoning, or simply a long answer for questions expecting such an answer. The topic field is solely dependent on the available information of the dataset. Ideally, it contains a hierarchical list of domains and subdomains for each question.

Question import is in most cases done by mapping variously named data fields of the original datasets to the corresponding fields in our internal data structure. In rare cases, the original data needs to be processed with pattern matching techniques to split the data within a single field in the original data structure to obtain data for multiple fields in the target structure. We will also release these semi-automated processing scripts together with all converted Q&A datasets upon publication of this paper.

7.2. LLM Querying

In cases, where given answers by a specific LLM Q&A dataset combination, are not available, we need to obtain the answers for each question for each LLM to be evaluated - which happens in an automated manner. In between each question, possible bias stemming from past requests must be avoided, which happens naturally when using LLM APIs, which do not store state between API requests. In this case, we have used OpenRouter (`https://openrouter.ai/`) platform, that offers the users to buy credits that are then used to query models. The process is simple, we have a Python script that reads the question set, and, for each question, we build a prompt, that instructs the system to answer in a certain format, and issue a call to OpenRouter with the corresponding API key. Then, the result of the prompt is processed to check whether the answer has been responded correctly. The results are stored as JSON files and then processed to obtain the desired input format. In cases where desired, additional information about each question can be acquired similarly, e.g., by prompting the LLM to be evaluated to provide a self-assessed difficulty rating.

7.3. Result Exploring

To ensure that LLMVisExplorer can be used across diverse contextsfrom scientific publications to interactive web applications-we designed the system around SVG-based rendering, which offers scalability, portability, and high-quality export suitable for both static and dynamic media. The core layout and visualization logic is implemented in Python, allowing machine learning practitioners to easily generate and configure visualizations without requiring web development expertise. While this choice sacrifices some interactive flexibility compared to browser-native libraries like D3.js, it prioritizes usability, reproducibility, and script-based integration with model evaluation pipelines.

The generated output consists of a self-contained HTML document, which embeds the SVG visualization alongside lightweight JavaScript components to enable interaction. The GUI features include our zoomable and clickable SVG visualization, the question-aware hierarchy editor built using jQuery and the TreeEditor plugin, and tab-based navigation for switching between editing and visualization views. This design supports standalone usage, sharing, and publication, while remaining extensible for downstream applications.

8. Use Cases

To demonstrate the expressivity of the visualizations generated with LLMVisExplorer, we demonstrate it on several Q&A benchmark LLM combinations. We will discuss first single-model results, before discussing multi-model results.

Single-model results. Fig. 1 shows the results of GPT-3.5 on PubMedQA (large-scale version in supplement). As can be seen,

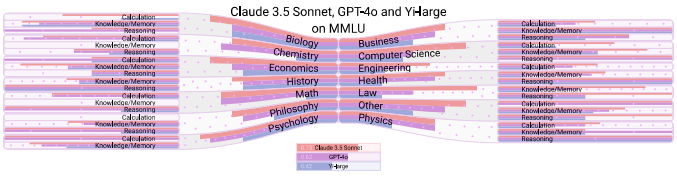


Fig. 7: Claude 3.5 Sonnet, GPT-4o, and Yi-large performance on a 57 subdomain subset of MMLU - one of the leading LLM benchmarks.

[1] `https://github.com/ad-freiburg/large-qa-datasets`

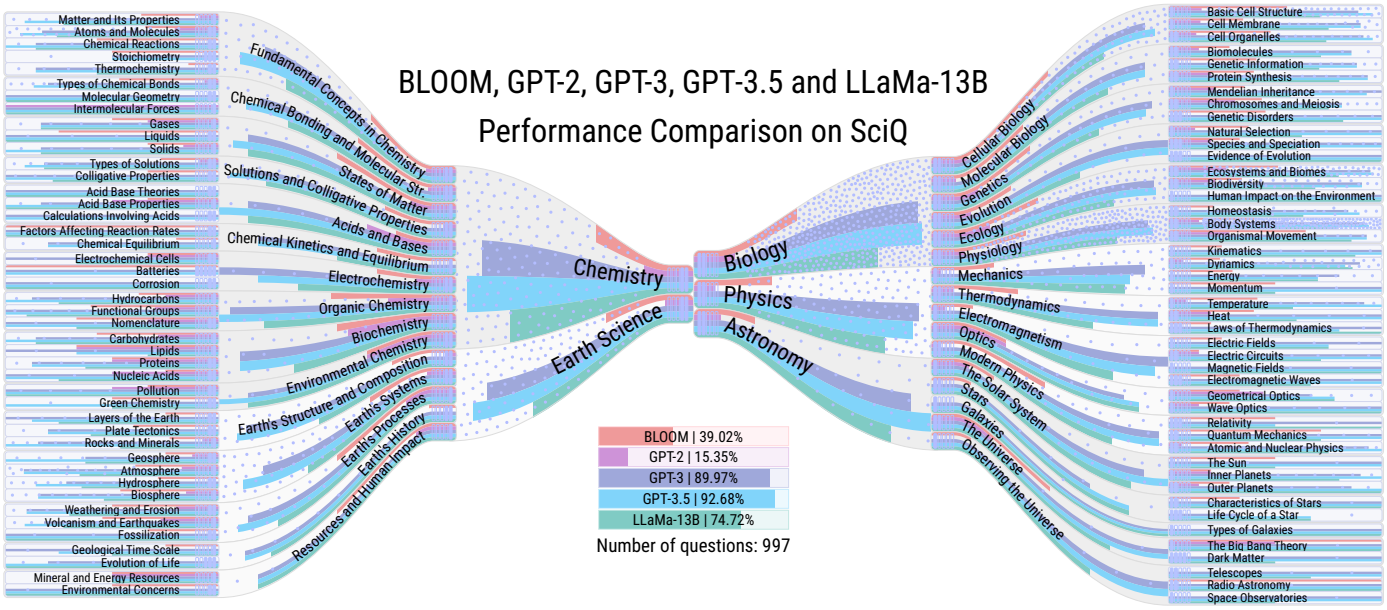


Fig. 8: Comparison of BLOOM, GPT-2, GPT-3,and LLaMa-13B on the SciQ natural sciences Q&A test set. Bars show model accuracy, blue noise number of questions, and discrete progress bar icons model-agnostic difficulty rating - each aggregated per knowledge hierarchy level.

GPT-3.5 performs vastly different for different subdomains, ranging from 59% to 78% on the first level, hinting that especially for genetics, regenerative medicine and infectious diseases, additional refinements would be required. While these extremes might raise concern on question distribution, LLMVisExplorer enables us to evaluate this as well through the blue noise display of the respective population. As such, it becomes apparent that some leaf nodes are only populated with one or two samples, which renders the respective findings in these areas statistically unreliable.

Fig. 6 shows GPT-3.5's results on the US Multistate Bar Examination questions. We believe that this is a good example for a knowledge domain in which LLMVisExplorer helps to formulate a justified distrust in GPT-3.5. As such, *Estates and Future Interests* under *Real Property* show a relatively dense population, while also showing a rather low accuracy.

Multi-model results. Fig. 7 shows a multi-model visualization of Claude 3.5 Sonnet, GPT-4o, and Yi-Large across a 57 subdomain subset of the MMLU benchmark. While Claude 3.5 Sonnet performed best on the selected subdomains, GPT-4o also showed consistent top-tier performance across many subdomains. Claude 3.5 Sonnet outperforms GPT-4o on almost all subdomains, except Math where they are on par and Chemistry, where GPT-4o is ahead. Interestingly, Yi-Large exhibited competitive results in Biology, but performed subpar in Chemistry - which can be considered related. We believe that this comparison showcases how LLMVisExplorer not only highlights general capabilities, but also uncovers specialized strengths and potential blind spots across models - critical for informed model selection in domain-specific deployments.

Fig. 8 shows the comparison of multiple models on the test split of the SciQ dataset. The first observation we can make is about the general performance, where GPT-3.5 and GPT-3 perform best, but also LLaMa-13B performs surprisingly well given its relatively small size (13B vs. 175B parameters in GPT-3.5). The subdomain visualization further exposes that GPT-3.5 performs quite different on different subdomains. However, the low density of the blue noise dots in some of these subdomains, signals caution as these subdomains contain very few questions, e.g., Astronomy. So we can conclude, that while GPT-3.5 scores best overall, is not for each subdomain the best scoring model. Most notably, and with a rather high significance, this is true for Physics, where GPT-3 is not only better on average across the entire field, but even for the next subdomain level is on par or better in all but one case than its successor model GPT-3.5.

9. Evaluation

Our evaluation has three complementary goals. First, we assess a specific perceptual trade-off of the proposed TreeVis design through a controlled study (Sec. 9.1). Second, we gather qualitative feedback on the usefulness and interpretability of LLMVisExplorer from domain experts and practitioners (Sec. 9.2). Third, we demonstrate the applicability of the system across representative models and benchmark datasets through additional usage scenarios (Sec. 9.3). Together, these components are intended to characterize both a focused perceptual property of the visualization and its broader practical utility, while not constituting a comprehensive comparison against all alternative hierarchy visualization techniques; in particular, the controlled study isolates the effect of embedding the primary metric into curved hierarchical regions through comparison with a closely related bar-based alternative.

9.1. Controlled Experiment

The experiment followed a randomized within-subject design with 23 participants. Each participant completed both visualization conditions, *TreeVis* and *BarVis*, with the order counterbalanced across two groups to reduce ordering effects. For each

condition, participants made 15 visual estimates of percentage bar length: five estimates for top-level nodes and ten estimates for mid-level nodes. Participants inspected the visualizations in fullscreen, wrote their estimates on paper, and were not given a time limit. No training trials were provided, because the goal was to assess how intuitively participants could interpret the visual encodings without prior exposure. Participants completed the study independently on their own devices. All participants had a computer-science background, ranging from senior undergraduate students to PhD candidates and postdoctoral rsearchers. Further details on the synthetic data generation procedure and raw judgment errors are provided in the supplementary material (Sec. 4).

For each participant and condition, we computed the average absolute error by taking the absolute difference between the participant's estimate and the ground truth, then averaging across the available judgments. In the *BarVis* condition, the mean absolute error was approximately 2.30% points. In contrast, the *TreeVis* condition yielded a mean absolute error of about 3.72% points.

A paired t-test comparing the average errors between conditions indicated that the observed difference is significant ($t(22) = 2.11$, $p = 0.046$). These results indicate that the straighter *BarVis* alternative can be advantageous for isolated bar-length estimation. However, this experiment intentionally evaluates only one low-level perceptual task and does not capture the broader analytical objective of LLMVisExplorer, namely the integrated exploration of hierarchical structure, local context, and performance variation within a single view.

Despite this statistical significance, the overall magnitude of error in both conditions remains modest. We therefore interpret this result as a perceptual trade-off of the *TreeVis* design: compared with *BarVis*, it reduces precision for isolated value judgments, but supports a more integrated representation of hierarchy and performance. In addition, LLMVisExplorer provides exact numerical values through labels and interactive tooltips when precise lookup is required. We chose BarVis as the comparison condition because it preserves much of the same hierarchical organization while removing the curved embedding of the primary metric, thereby allowing us to isolate the perceptual effect of curvature more directly than would be possible with structurally more different alternatives such as treemaps. Accordingly, the results should not be interpreted as establishing overall analytical superiority of either design, but rather as characterizing one specific perceptual trade-off within the broader visualization design space.

To complement the controlled experiment on isolated bar-length estimation, we conducted a small exploratory follow-up with 12 participants recruited separately from the main study, focused on participants' final comparison judgments between *TreeVis* and *BarVis* after both had completed a short set of structural lookup tasks (e.g., identifying node membership and parent-child relations) under each condition. Further details on the study protocol and questionnaire are provided in the supplementary material (Sec. 4). This follow-up was intended to probe perceived support for hierarchy-aware interpretation rather than low-level magnitude estimation; we report only the final comparison judgments here, as these lookup tasks were not designed or powered as a controlled accuracy comparison.

Across 12 responses, participants were mainly computer science students ranging from bachelor's to master's level. *TreeVis* was selected more often as making parent-child relationships easier to understand (83.3%), while *BarVis* was selected by 16.7% of participants. For identifying weak regions, responses were mixed: *TreeVis* and *BarVis* were each selected by 41.7% of participants, with 16.7% reporting no difference. On the third overall comparison item, *TreeVis* was again selected more often (66.7%), compared with *BarVis* (16.7%) and no difference (16.7%).

These results should be interpreted cautiously: they reflect subjective preference after brief exposure, not measured analytical performance on realistic weakness-finding or model-comparison tasks. Nevertheless, they are consistent with the design trade-off identified above - *BarVis* can support more accurate isolated bar-length judgments, whereas *TreeVis* appears to provide stronger perceived support for understanding hierarchical structure and parent-child relations. The mixed responses for identifying weak regions further suggest that neither design is uniformly preferable across all tasks. Establishing whether this perceived structural support translates into measurable analytical performance gains remains an open question for future, task-based evaluation.

9.2. LLM Practitioner Evaluation

To assess the interpretability of our visualization layout, we conducted an informal, semi-structured interview session with three domain experts in the medical field, specifically from radiology, nuclear medicine, and medical radiation physics, which we have exposed to an early realization of our visualization. The participants had between 11 and 27 years of medical research experience. While all three were generally familiar with LLMs and ChatGPT in particular, two had prior experience using ChatGPT in medical contexts and expressed both curiosity and caution regarding its factual accuracy.

The interview focused on evaluating GPT-3.5's performance on the PubMedQA dataset (see Fig. 1). Since none of the participants were previously aware of this dataset, we first provided a brief explanation of its structure, question types, and annotation scheme. The experts welcomed the availability of such a dataset and were interested in exploring the model's capabilities across different medical subdomains.

The visualization encouraged exploratory behavior. Each expert engaged with subdomains related to their areas of research and commented on the coverage of medical topics. When asked to assess the underlying knowledge hierarchy, participants identified several relevant subdomains - such as orthopedics, dermatology, ophthalmology, and internal medicine-that appeared underrepresented or missing. Although, they did not find any clearly erroneous entries, one participant questioned the placement of "Computed Tomography (CT) Imaging" as a subdomain of "Radiographic Imaging." After a brief discussion, they agreed that the categorization could be interpreted in multiple valid ways. All three participants were able to interpret the visualization after a brief (<1 minute) explanation. They successfully

extracted subdomain-level accuracy values, identified areas of strength and weakness, and appreciated the inclusion of question density as a visual cue. Notably, one expert correctly inferred from the visualization that ethical questions were less frequent in the dataset. This feedback confirms that the visual encoding is accessible to domain experts without requiring prior visualization training.

Overall, the interviews indicated a clear interest among medical researchers in better understanding the domain-specific capabilities of LLMs. Participants considered our visualization a valuable tool for this purpose, particularly due to its ability to expose performance gaps in an interpretable and interactive manner. The decoupling of hierarchy generation and visualization was considered a strength, offering the flexibility to tailor hierarchies to specific use cases. One participant expressed interest in applying the system to a broader dataset and customizing the hierarchy to match their research focus.

9.3. LLM Expert Evaluation

We conducted structured interviews with nine experts from the computer vision and visualization communities. The participants were academic researchers, including professors, with experience in LLM development, evaluation, and/or the use of LLMs in their research. Participants were recruited through our academic network based on relevant expertise in visualization, computer vision, and the use or evaluation of LLMs. Each interview lasted approximately 20-30 minutes. Participants reviewed LLMVisExplorer using visualizations based on the SciQ and PubMedQA datasets, including both single-model and multi-model comparison scenarios. The structured interview covered the comprehensibility of the visualization, its usefulness for identifying model strengths and weaknesses, the usefulness of the hierarchy and additional visual encodings, and potential improvements. Participants provided feedback through rating-scale questions and free-text responses.

The responses were summarized descriptively according to the main topics covered by the structured interview. Rating-scale responses were used to characterize overall response tendencies, while free-text responses were examined for recurring observations, preferences, and concerns. We report both favorable and critical feedback; no formal qualitative coding or thematic analysis was performed. The purpose of this evaluation was therefore to collect exploratory expert feedback on the perceived usefulness and interpretability of LLMVisExplorer rather than to measure objective task performance.

All participants considered LLMVisExplorer more helpful for analyzing model performance than traditional aggregate accuracy displays or confusion matrices. They reported that the visual breakdown by subdomains gave them a more nuanced understanding of where the model excelled or struggled. For instance, one participant noted that the visualization provided context that an overall accuracy metric alone could not, emphasizing how it helped them quickly pinpoint relative strengths and weaknesses across different subdomains. This sentiment was reflected in the rating feedback as well, with participants giving high marks for the LLMVisExplorer's usefulness compared to conventional metrics. Participants agreed that LLMVisExplorer was useful for identifying weak subdomains in the model's knowledge. By examining the knowledge hierarchy, they could easily spot specific subdomains where the model's accuracy was notably low. This capability was highlighted as a major advantage: several participants mentioned they might have overlooked these weaknesses if they had only seen an average score. None of the participants found the visualization itself misleading; in fact, they felt it reduced the chance of being misled by aggregate metrics. A couple of participants did caution, however, that small sample sizes in certain subtopics could make those performance metrics less reliable, though showing the question counts per node helped them keep this in perspective.

The feedback indicated that participants would use insights from LLMVisExplorer to support model improvement. They explained that by revealing concrete weak areas, the tool could guide targeted data augmentation or fine-tuning. For example, if the visualization showed the model underperforming in a specific category, they could gather additional training examples for that subtopic or adjust the model accordingly. Participants noted that this targeted approach to error analysis was more actionable than working from an overall accuracy alone. Some also appreciated the multi-model comparison view, saying it helped them determine which model's approach was superior in certain areas or which model required further improvement.

Opinions differed on whether empty leaf nodes (topics with no questions) should be displayed in the hierarchy. Several participants preferred to hide empty nodes, arguing that showing categories with no data could clutter the view and confuse users about the model's performance in those areas. They felt that if no questions were present for a given subdomain, it was clearer to simply omit that branch from the visualization. In contrast, other participants wanted all predefined subdomains to be shown regardless of data coverage, as this provides context about the scope of the dataset and reveals which subdomains were not tested at all. They believed that even empty subdomains carried informative value (indicating potential gaps in the evaluation), and suggested perhaps visually deemphasizing them rather than removing them entirely.

Participants also gave specific feedback on individual visual elements. While they generally appreciated having indicators of question difficulty and potential hallucination, several were uncertain about how to interpret these without additional explanation. It was unclear to some whether the difficulty level was an inherent property of the question or derived from model performance, and what exactly constituted a hallucination in this context (since SciQ and PubMedQA provides factual answers). They suggested that clearer definitions or on-screen explanations for these indicators would be helpful. Additionally, a few participants felt that the color scheme lacked sufficient contrast to distinguish certain elements clearly (such as correct vs. incorrect answers or varying difficulty levels), which made parts of the visualization challenging to read for some users. Despite these issues, which we have addressed with legends and higher contrast colors after the study, participants were generally in favor of surfacing such information (e.g., flagging if a model's

answer was likely unsupported or if a question was particularly challenging), as long as the presentation was clear and unambiguous. Overall, the feedback from the nine participants was predominantly favorable regarding the comprehensibility and perceived usefulness of LLMVisExplorer, while also identifying limitations related to visual clarity, interpretation of secondary encodings, dataset coverage, and the presentation of empty subdomains.

Limitations. LLMVisExplorer is intended to support an integrated, hierarchy-aware overview of LLM performance across complex knowledge domains. By combining hierarchical structure, primary performance values, question-density cues, and optional secondary metrics, it helps users identify subdomain-level strengths and weaknesses that may be hidden by aggregate benchmark scores. At the same time, these design choices introduce several trade-offs and limitations.

First, the curved embedding of the primary metric supports a compact and hierarchy-aware overview, but reduces the precision of isolated value judgments compared with straighter bar-based alternatives. Second, compact leaf stacking improves space efficiency at the cost of strict depth regularity at the leaf level. Third, secondary icon encodings can lose visibility in dense static views, especially in multi-model color settings. More generally, combining curved bars, icons, and dot-density cues within one compact view may increase cognitive load relative to simpler single-metric displays, particularly for first-time users. The secondary encodings can therefore be selectively disabled when they are not relevant to the current task. Our expert feedback suggests that these encodings become easier to interpret after brief explanation, but we did not measure cognitive load or learning effects directly. Fourth, very large hierarchies may exceed the perceptual capacity of a single overview and therefore require progressive exploration through filtering, collapsing, or selective metric display. Fifth, our follow-up comparison judgments provide only exploratory evidence because the sample is small and not designed as a fully powered comparative usability study. Finally, the semi-automated hierarchy construction workflow provides an initial structured organization rather than validated ground truth, and may require user refinement depending on domain and prompting setup.

10. Conclusions and Future Work

We introduced LLMVisExplorer, an interactive visualization system designed to support multifaceted, subdomain-aware evaluations of LLMs on Q&A datasets. By integrating hierarchical knowledge layouts, extendable metric encodings, and interaction techniques grounded in visualization principles, LLMVisExplorer helps surface insights that are often obscured by single-number evaluations. LLMVisExplorer supports both model developers and users in identifying strengths, weaknesses, and blind spots, facilitating more responsible deployment and red-teaming of LLMs. Alternative visualization designs could have been chosen for this task, including icicle-style layouts with separate quantitative tracks, treemap variants with detached metric panels, small-multiple summaries, or matrix-based views for fine-grained leaf comparison. These alternatives offer different trade-offs in structural continuity, compactness, labeling, and quantitative precision. We chose the present design because it keeps hierarchical grouping, primary performance encoding, and optional secondary/contextual annotations visually integrated within one overview while still supporting drill-down to question-level evidence. The resulting system realizes our design requirements for hierarchical, multi-metric, comparative, and interactive analysis, while the controlled experiment and expert evaluations provide supporting evidence for its interpretability and perceived usefulness. We therefore regard the proposed design as a deliberate choice within the broader hierarchy-visualization design space.

Because the visual analysis is mediated through the induced hierarchy, any noise, bias, or instability in the hierarchy construction can affect which subdomains appear salient and which weaknesses appear localized. The resulting structure should therefore be interpreted as an editable analytical scaffold rather than a ground-truth ontology. An important direction for future work is to evaluate how stable the observed performance patterns remain under plausible alternative hierarchy constructions, labeling choices, or manually refined taxonomies.

In the future, we plan to extend LLMVisExplorer along several dimensions. First, we plan to close the loop between the interactive Tree Editor and the visualization itself, so that confirming an edit triggers in-browser recomputation of all affected summaries and allows the edited hierarchy to be exported and reloaded for later reuse. Second, we aim to support interactive comparison across temporal model updates, enabling longitudinal performance tracking. Third, we envision tight integration with automated error categorization and explanation techniques, to better contextualize failure cases. Finally, we will investigate how user-generated hierarchies and feedback can improve test set construction and benchmarking coverage across diverse domains. We hope that LLMVisExplorer encourages broader adoption of structured, interpretable LLM evaluation in both research and applied settings.

Acknowledgments

References

[1] Anderson, LW, Krathwohl, DR, Airasian, PW, Cruikshank, KA, Mayer, RE, Pintrich, PR, et al. A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives, abridged edition. Pearson; 2000.

[2] Arawjo, I, Swoopes, C, Vaithilingam, P, Wattenberg, M, Glassman, E. ChainForge: A visual toolkit for prompt engineering and LLM hypothesis testing. arXiv preprint arXiv:230909128 2023;.

[3] Brinkmann, A. Graphical knowledge display–mind mapping and concept mapping as efficient tools in mathematics education. Mathematics Education Review 2003;16(4):35–48.

[4] Brown, T, Mann, B, Ryder, N, Subbiah, M, Kaplan, JD, Dhariwal, P, et al. Language models are few-shot learners. Advances in neural information processing systems 2020;33:1877–1901.

[5] Bubeck, S, Chandrasekaran, V, Eldan, R, Gehrke, J, Horvitz, E, Kamar, E, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. 2023. `arXiv:2303.12712`.

[6] Burch, M, Weiskopf, D. On the benefits and drawbacks of radial diagrams. In: Handbook of human centric visualization. Springer; 2013, p. 429–451.

[7] Chang, Y, Wang, X, Wang, J, Wu, Y, Yang, L, Zhu, K, et al. A survey on evaluation of large language models. ACM Trans Intell Syst Technol 2024;15(3). URL: https://doi.org/10.1145/3641289. doi:10.1145/3641289.

[8] Cheng, F, Zouhar, V, Arora, S, Sachan, M, Strobelt, H, El-Assady, M. Relic: Investigating large language model responses using self-consistency. In: CHI Conference on Human Factors in Computing Systems. 2024, p. 1–18.

[9] Cheng, F, Zouhar, V, Chan, RSM, Fürst, D, Strobelt, H, El-Assady, M. Understanding large language model behaviors through interactive counterfactual generation and analysis. IEEE Transactions on Visualization and Computer Graphics 2026;32(1):846–856. doi:10.1109/TVCG.2025.3634646.

[10] Chiou, CC. The effect of concept mapping on students' learning achievements and interests. Innovations in Education and teaching International 2008;45(4):375–387.

[11] Choi, JH, Hickman, KE, Monahan, A, Schwarcz, D. ChatGPT goes to law school. Available at SSRN 2023;.

[12] Cleveland, WS. The Elements of Graphing Data. Wadsworth Publ. Co.; 1985.

[13] Crisan, A, Drouhard, M, Vig, J, Rajani, N. Interactive model cards: A human-centered approach to model documentation. In: Conference on Fairness, Accountability, and Transparency. 2022, p. 427–439.

[14] D'Antoni, AV, Zipp, GP, Olson, VG, Cahill, TF. Does the mind map learning strategy facilitate information retrieval and critical thinking in medical students? BMC medical education 2010;10:1–11.

[15] Desai, S, Durrett, G. Calibration of pre-trained transformers. EMNLP 2020;.

[16] Edwards, S, Cooper, N. Mind mapping as a teaching resource. The clinical teacher 2010;7(4):236–239.

[17] Hendrycks, D, Burns, C, Basart, S, Zou, A, Mazeika, M, Song, D, et al. Measuring massive multitask language understanding. arXiv preprint arXiv:200903300 2020;.

[18] Honovich, O, Aharoni, R, Herzig, J, Taitelbaum, H, Kukliansy, D, Cohen, V, et al. True: Re-evaluating factual consistency evaluation. NAACL 2022;.

[19] Jin, Q, Dhingra, B, Liu, Z, Cohen, W, Lu, X. PubMedQA: A dataset for biomedical research question answering. In: Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019, p. 2567–2577.

[20] Johnson, B, Shneiderman, B. Tree-maps: A space filling approach to the visualization of hierarchical information structures. Tech. Rep.; UM Computer Science Department; CS-TR-2657; 1998.

[21] Joshi, M, Choi, E, Weld, DS, Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In: Annual Meeting of the Association for Computational Linguistics. 2017,.

[22] Kahng, M, Tenney, I, Pushkarna, M, Liu, MX, Wexler, J, Reif, E, et al. Llm comparator: Interactive analysis of side-by-side evaluation of large language models. IEEE Transactions on Visualization and Computer Graphics 2025;31(1):503–513. URL: https://doi.org/10.1109/TVCG.2024.3456354. doi:10.1109/TVCG.2024.3456354.

[23] Khartabil, D, Collins, C, Wells, S, Bach, B, Kennedy, J. Design and evaluation of visualization techniques to facilitate argument exploration. In: Computer Graphics Forum; vol. 40. 2021, p. 447–465.

[24] Kruskal, JB, Landwehr, JM. Icicle plots: Better displays for hierarchical clustering. The American Statistician 1983;37(2):162–168.

[25] Kung, TH, Cheatham, M, Medenilla, A, Sillos, C, De Leon, L, Elepaño, C, et al. Performance of ChatGPT on USMLE: Potential for ai-assisted medical education using large language models. PLOS Digital Health 2023;2(2):e0000198.

[26] Liang, P, Bommasani, R, Lee, T, Tsipras, D, Soylu, D, Yasunaga, M, et al. Holistic evaluation of language models. 2022. arXiv:2211.09110.

[27] Liu, Y, Iter, D, Xu, Y, Wang, S, Xu, R, Zhu, C. G-eval: Nlg evaluation using gpt-4 with better human alignment, may 2023. arXiv preprint arXiv:230316634 2023;.

[28] Mahowald, K, Ivanova, AA, Blank, IA, Kanwisher, N, Tenenbaum, JB, Fedorenko, E. Dissociating language and thought in large language models: a cognitive perspective. arXiv preprint arXiv:230106627 2023;.

[29] Mitchell, M, Wu, S, Zaldivar, A, Barnes, P, Vasserman, L, Hutchinson, B, et al. Model cards for model reporting. In: Conference on fairness, accountability, and transparency. 2019, p. 220–229.

[30] Nitko, AJ. Educational assessment of students. Prentice-Hall, Inc.; 1996.

[31] Nori, H, King, N, McKinney, SM, Carignan, D, Horvitz, E. Capabilities of GPT-4 on medical challenge problems. arXiv preprint arXiv:230313375 2023;.

[32] Ondov, B, Jardine, N, Elmqvist, N, Franconeri, S. Face to face: Evaluating visual comparison. IEEE Transactions on Visualization and Computer Graphics 2018;25(1):861–871.

[33] OpenAI, . GPT-4 technical report. 2023. arXiv:2303.08774.

[34] Raji, ID, Smart, A, White, RN, Mitchell, M, Gebru, T, Hutchinson, B, et al. Closing the ai accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In: Conference on fairness, accountability, and transparency. 2020, p. 33–44.

[35] Rein, D, Hou, BL, Stickland, AC, Petty, J, Pang, RY, Dirani, J, et al. Gpqa: A graduate-level google-proof q&a benchmark. In: First Conference on Language Modeling. 2024,.

[36] Ribeiro, MT, Wu, T, Guestrin, C, Singh, S. Beyond accuracy: Behavioral testing of nlp models with checklist. ACL 2020;.

[37] Scao, TL, Fan, A, Akiki, C, Pavlick, E, Ilić, S, Hesslow, D, et al. BLOOM: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:221105100 2022;.

[38] Schulz, HJ. Treevis. net: A tree visualization reference. IEEE Computer Graphics and Applications 2011;31(6):11–15.

[39] Schulz, HJ, Hadlak, S, Schumann, H. The design space of implicit hierarchy visualization: A survey. IEEE Transactions on Visualization and Computer Graphics 2010;17(4):393–411.

[40] Selvi, RT, Chandramohan, G. Case study on effective use of mind map in engineering education. In: 2018 IEEE Tenth International Conference on Technology for Education (T4E). 2018, p. 205–207.

[41] Sevastjanova, R, Vogelbacher, S, Spitz, A, Keim, D, El-Assady, M. Visual comparison of text sequences generated by large language models. In: IEEE Visualization in Data Science (VDS). 2023, p. 11–20.

[42] Shneiderman, B. The eyes have it: A task by data type taxonomy for information visualizations. In: The craft of information visualization. Elsevier; 2003, p. 364–371.

[43] Shridhar, K, Macina, J, El-Assady, M, Sinha, T, Kapur, M, Sachan, M. Automatic generation of socratic subquestions for teaching math word problems. arXiv preprint arXiv:221112835 2022;.

[44] Spinner, T, Fürst, D, El-Assady, M. innspector: Visual, interactive deep model debugging. arXiv preprint arXiv:240717998 2024;.

[45] Spinner, T, Kehlbeck, R, Sevastjanova, R, Stähle, T, Keim, DA, Deussen, O, et al. generaitor: Tree-in-the-loop text generation for language model explainability and adaptation. ACM Transactions on Interactive Intelligent Systems 2024;14(2):1–32.

[46] Strobelt, H, Hoover, B, Satyanarayan, A, Gehrmann, S. LMdiff: A visual diff tool to compare language models. In: Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 2021,.

[47] Suskie, L. Assessing student learning: A common sense guide. John Wiley & Sons; 2018.

[48] Talmor, A, Berant, J. The web as a knowledge-base for answering complex questions. arXiv preprint arXiv:180306643 2018;.

[49] Tamkin, A, Brundage, M, Clark, J, Ganguli, D. Understanding the capabilities, limitations, and societal impact of large language models. arXiv preprint arXiv:210202503 2021;.

[50] Taori, R, Gulrajani, I, Zhang, T, Dubois, Y, Li, X, Guestrin, C, et al. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca; 2023.

[51] Touvron, H, Lavril, T, Izacard, G, Martinet, X, Lachaux, MA, Lacroix, T, et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:230213971 2023;.

[52] Tsatsaronis, G, Balikas, G, Malakasiotis, P, Partalas, I, Zschunke, M, Alvers, MR, et al. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. BMC bioinformatics 2015;16:1–28.

[53] van Onzenoodt, C, Singh, G, Ropinski, T, Ritschel, T. Blue noise plots. In: Computer Graphics Forum; vol. 40. 2021, p. 425–433.

[54] Wang, A, Singh, A, Michael, J, Hill, F, Levy, O, Bowman, SR. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. 2018, p. 353–355.

[55] Wang, Y, Ma, X, Zhang, G, Ni, Y, Chandra, A, Guo, S, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In: Neural Information Processing Systems Datasets and

Benchmarks Track. 2024,.
[56] Welbl, J, Stenetorp, P, Riedel, S. Constructing datasets for multi-hop reading comprehension across documents. Transactions of the Association for Computational Linguistics 2018;6:287–302.
[57] White, C, Dooley, S, Roberts, M, Pal, A, Feuer, B, Jain, S, et al. Livebench: A challenging, contamination-free LLM benchmark. ICLR 2025 2025;.
[58] Yang, Z, Qi, P, Zhang, S, Bengio, Y, Cohen, WW, Salakhutdinov, R, et al. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. arXiv preprint arXiv:180909600 2018;.
[59] Yin, M, Wortman Vaughan, J, Wallach, H. Understanding the effect of accuracy on trust in machine learning models. In: CHI Conference on Human Factors in Computing Systems. 2019, p. 1–12.
[60] Zhan, X, Goyal, A, Chen, Y, Chandrasekharan, E, Saha, K. SLM-Mod: Small language models surpass LLMs at content moderation. In: NAACL. 2025,.
[61] Zhao, Z, Wallace, E, Feng, S, Klein, D, Singh, S. Calibrate before use: Improving few-shot performance of language models. In: International conference on machine learning. 2021, p. 12697–12706.
[62] Zheng, L, Chiang, WL, Sheng, Y, Zhuang, S, Wu, Z, Zhuang, Y, et al. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In: NeurIPS Datasets and Benchmarks Track. 2023,.