Linting is People! Exploring the Potential of Human Computation as a Sociotechnical Linter of Data Visualizations

Anamaria Crisan
Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
amcrisan@uwaterloo.ca

Andrew M McNutt
Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, Utah, USA
andrew.mcnutt@utah.edu

Abstract

Traditionally, linters are code analysis tools that help developers by flagging potential issues from syntax and logic errors to enforcing syntactical and stylistic conventions. Recently, linting has been taken as an interface metaphor, allowing it to be extended to more complex inputs, such as visualizations, which demand a broader perspective and alternative approach to evaluation. We explore a further extended consideration of linting inputs, and modes of evaluation, across the puritanical, neutral, and rebellious dimensions. We specifically investigate the potential for leveraging human computation in linting operations through Community Notes—crowd-sourced contextual text snippets aimed at checking and critiquing potentially accurate or misleading content on social media. We demonstrate that human-powered assessments not only identify misleading or error-prone visualizations but that integrating human computation enhances traditional linting by offering social insights. As is required these days, we consider the implications of building linters powered by Artificial Intelligence.

CCS Concepts

• **Human-centered computing** → **Visualization**; **Interaction paradigms**; • **Information systems** → **Crowdsourcing**.

Keywords

Community Notes, human-powered computing, linting, provocation

ACM Reference Format:
Anamaria Crisan and Andrew M McNutt. 2025. Linting is People! Exploring the Potential of Human Computation as a Sociotechnical Linter of Data Visualizations. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3706599.3716230

1 Introduction

Linters are a means of collaboratively establishing correctness over a digitally analyzable object in a standardized and enforced manner. Traditionally, linters have been used exclusively in the domain of code, acting as a sort of digital spell checker for errors that can be identified through static analysis. For instance, identifying stylistic issues (like missing semicolons), functional issues (such as undefined variables), and domain-specific issues (such as certain operations being slow [71]). Recent work has explored generalizing the idea of a linter out of code and into other domains, such as visualization [10], spreadsheets [4], and color palettes [46].

Crucially, linters are a useful tool when the expression domain has been widened enough that mistakes can be of consequence (cf. Figure 1). There are, naturally, two areas in which this might apply: how an idea was expressed (e.g. the sentence "is squeak doodads when altercation" is non-syntactically valid) and the content of the expression (e.g. the sentence "colorless green ideas sleep furiously" is famously syntactically valid but semantically meaningless [63]). In prior work [46] we argued that, as analysis tools improve, the most useful approach for linters is to analyze the latter of these. That is, to provide useful operationalization of knowledge about the execution domain rather than the input plane.

Yet! It is often not clear what it means for something to be right. Notions of correctness, particularly in sociotechnical tools (like visualizations), are ambiguous in general, and community- and usage-specific in particular. For instance, the rainbow color map is commonly argued [7] in the visualization community as being something to be avoided (as it can cause hallucinations about the data via steep perceptual cliffs and so on). Yet! In some communities, particularly in weather [60, 61], using standard color ramps of this format is nearly mandatory. We suggest that—in interacting with scientists working in weather-adjacent areas—this is in part because using it indicates that you are a member of the in-group that you are trying to communicate with—but also because the people in that community are trained and familiar with reading information presented in that fashion. Another example: the difficulty of ski runs in North America is delineated via a scale (●-■-◆). While this is broadly understood (and because of the double encoding of shape and color is color vision deficiency friendly), it generally does not follow conventions in perceptually comprehensible sequential ordering. Evidently, cultural convention can win out over scientifically motivated usage! This highlights the boundaries of scientifically motivated linters: cultural knowledge can be hard to capture in simple computer-based checking tools, highlighting that there may be a need to explore other means of running a linter.

In this work we do just that: we extend this idea of linters being socially positioned to cover, not just the usage of lint, but their execution as well. We consider the potential for leveraging human computation as a means of linting data visualizations to surface instances of valid visualizations (e.g. adhering to established design guidelines), but also to assess whether it is meaningful to the intended audience. Human computation leverages groups of people

CHI EA '25, Yokohama, Japan

ACM ISBN 979-8-4007-1395-8/25/04
https://doi.org/10.1145/3706599.3716230

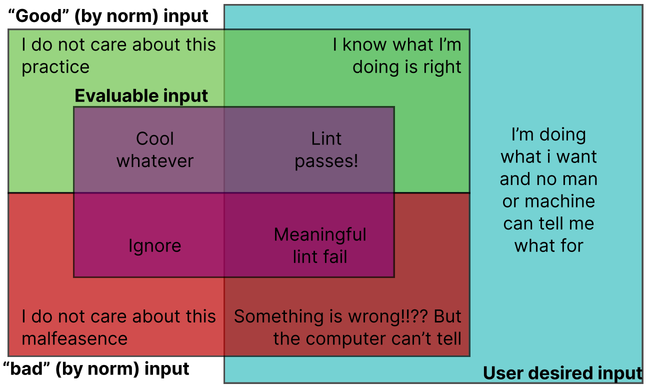

"Good" (by norm) input
I do not care about this practice
I know what I'm doing is right
Evaluable input
Cool whatever
Lint passes!
I'm doing what i want and no man or machine can tell me what for
Ignore
Meaningful lint fail
I do not care about this malfeasence
Something is wrong!!?? But the computer can't tell
"bad" (by norm) input
User desired input

Figure 1: A characterization of the input space for linting as Euler diagram. Linters tend to exist in domains where the input space is expressive enough for there to be types of input that are not correct by default. These rich domains are common, for instance, language, text, visualization, film, and many other mediums support such expansive input. Within these spaces linters can only reason about a subset of possible input (which is itself only a subset of the space for which there are community norms), which is possibly intersecting the space of input the user wishes to make.

to perform computational tasks, for example, data labeling, but has not been previously explored in the context of linting.

To do so we reflect on Community Notes (née Birdwatch) [57], which is a crowd-sourced moderation program that allows people to add context about potentially misleading content appearing on social media. While Community Notes serve a variety of purposes, we consider specific instances of their use to "lint" data visualizations (Figure 3). To explore the implications of linting as a metaphor, we reflect on its applications across the dimensions of its inputs and evaluative methods. Beginning with traditional code linters, which we consider to represent puritanical ideas of linting inputs and evaluative methods, we extend our exploration to more neutral and even rebellious perspectives.

2 Related Work

This work draws on linting (and related abstractions), as well as the idea of human computation.

2.1 Linting and Similar Abstractions

A key idea in this work is that linters are sociotechnical mediators of community standards. A good example of this can be seen in linters that operate over documentation. The earliest of these that we are aware of, write-good [19], offers lint-style feedback on effective writing in markdown files (capturing issues like inappropriate use of passive voice or turns of phrase). The Inclusive Language Plugin [17] for the static site generator Eleventy automatically pushes users to use inclusive language (avoiding clauses like "of course" or "just"). HaTe Detector [65] guides documentation authors away from using hateful terms, such as master/slave instead of primary/replica. Textlets [22] includes functionality that supports automated checking of adherence to a small collection of invariants within the particular domain of legal writing. The notion that linters are socially positioned objects is not new. For instance, the documentation of SQL Fluff [14] explicitly highlights that linters are a means to achieve communal adherence to a single style of authoring SQL queries. We emphasize that by automating correctness, linters act as a form of policy enforcement.

Closely associated with the notion of linters, is the notion of standards. Jackson and Barbrow [30] explore the meaning of standards and highlight how they can evolve over time and how standards need to be tuned to their local context (even when part of a larger project). A crucial part of standards is knowing when and how they can be broken appropriately. We see this as being closely aligned with the interface philosophy of linters: they offer a clear signal of correctness from a particular perspective, but can be altered or ignored to fit a specific situation.

Moreover, a key facet of using linters as an interface metaphor is that, for a given domain, there is a collection of best practices or domain knowledge that needs to be operationalized (i.e. a standard exists). This is a component of what differentiates them from a test suite: a linter is for a domain of work, whereas a test suite is for a project within a domain. An additional assumption in comparing linters and test suites is that linters are quick while test suites are slow. While this tension is worth exploring, in this work we forgo this component of this metaphor—instead focusing on the evaluatory and input aspects.

Within visualization there have been a number of different linting or lint-like systems. In earlier work McNutt et al. [46] survey these linters. These include tools for checking chart usage based on a notion of best practices [3, 43, 72], as well as more specific domains within visualization including exploratory modeling [31], choropleths [39], as well as the design of lint-style feedback [26]. Hull et al. [27] develop a system for automatically grading visualizations created in an online course. Connecting with documentation-centric linters, LitVis [66] includes a notion of linting over the visualization design process, requiring the designer to provide documentation (via markdown comments) that follows particular patterns (such as a Socratic dialog or consideration of data feminist principles). Sultanum et al. [56] propose a notion of data guards, automated checks for fostering trust in data artifacts (such as visualizations). This view of trust as being a key mediating factor in the acceptance of data artifacts is closely related to our perspective on linters being devices that mediate community standards, in that both situate technical perspectives on manipulation of sociotechnical behaviors. Kosara [35] criticizes the idea of linting visualizations, suggesting that being critical without offering solutions is not a helpful usage paradigm (although some linters do exactly that).

Some approaches offer linter-style feedback but with a different pose relative to the development or analysis cycle. Lisnic et al. [40] develop guard rails for preventing cherry-picking data. Narechania et al. [49] construct a data exploration system that highlights areas in data space not yet considered, surfacing potential biases.

Also closely related to linters is the question of teaching code quality. Keuning et al. [33] survey code quality in education, finding that some systems use linter or lint-like systems. For instance, McNutt et al. [45] conduct a study of creative coding editing tools with high school and college students, including a code-based linter. They find that students value linters as a means to internalize feedback. We emphasize that this type of automation can lead to automation biases—*if, as a beginner, the machine is telling me this is wrong then it's probably wrong, right?* This can be beneficial if the goal is to teach, but can be detrimental when the goal is a wrought artifact. Automation bias, of which this is an example, is a problem that plagues many linters [10, 39], not just those related to teaching.

Outside of visualization and text analysis, linters have been applied in a few different areas. Wang et al.'s [59] Farsight provide lint-style warnings when a prompt for an AI system might lead to harm. Data Linter [29] checks to ensure that machine learning data does not have determinant components, such as NaNs. ExceLint [4] uses entropic analysis to identify likely errors in spreadsheets. There are many areas and venues in which lint-style feedback can be applied left to explore. In this work we explore them in Community Notes.

2.2 Human Computation

Human computation is a widely used strategy that leverages collective human expertise and capabilities to complete tasks that are difficult for an automated method (e.g. AI agent) to perform [53]. Modern approaches use crowd-sourcing platforms (e.g. Mechanical Turk, Prolific) to distribute a micro task among many people and aggregate the results to formulate a final response [36]. Examples of human computation include dataset annotation and labeling [9, 55], document editing [5], identifying accessibility issues [23], or even visual design critiques [41, 70]. Response quality is an important factor, especially for complex tasks. Prior applications have shown that even non-expert workers can provide meaningful feedback to domain experts for complex tasks [41, 70]. However, there are a number of factors related to both the worker and the task itself [2]; to a reasonable extent these can be addressed through the design of applications using human computation.

While numerous visualization research studies have used crowdsourcing for conducting experiments (e.g. [6, 25]), leveraging human computation in visualization-specific tasks has been explored relatively less. For example, Willett et al. [64] leverage human computation to generate explanations for data visualizations. Wu et al. [67] uses human computation to learn the effective layouts of charts. The application of human computation to linting tasks more generally, and to visualization linting specifically, has not been explored. Yet this may be an intriguing avenue, given the breadth of applications where human computation has been applied and the overlap these approaches have with broad goals of linting.

A key consideration for leveraging human computation is to design an environment and method to manage crowd workers and tasks [2]. While the majority of human computation approaches primarily leverage crowd-working platforms (such as Prolific or Amazon Mechanical Turk) to do so, other avenues exist. For example, Moghadam et al. [48] used a Massive Open Online Course (MOOC) platform to gather and analyze a corpus of code examples and develop a prototype that assists students by providing targeted feedback. Even more prominently [37] CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) uses human computation as biomarker, but the resulting datasets collected from tasks are used to train models. More naturalistic scenarios for human computation, for example via social networks or other forms of human organization, have not been extensively explored or discussed. One exception is LabintheWild [54] which mixes a notion of social challenge with study incentive to complete different human-computation driven tasks broadly for scientific purposes. This rarity may be because it is neither intuitive nor natural to think of such organic interactions as a robust basis for reliable computational results. Yet, in the modern day, technology is closely woven into human interactions, whether they occur in person or online, and it is entirely possible for organic instances of even complex human computation to exist.

We consider a naturalistic application of human computation via social networks. We examine Community Notes and explore how they constitute both linting and human computation, setting aside (momentarily) questions of quality.

3 Abstracting Linting

Here we offer a categorization of the linting design space, lensed through the way in which the lint is evaluated and the context in which it is used. Following a common meme format [34], we organize these axes along measures of adherence to orthodoxy (ranging from purist to rebel). On one axis we consider the notion that linters are something that uses static analysis (i.e. code is not executed) to evaluate code (which is typical for code linters or compilers). On the other, we consider the means by which the specific target of analysis. We show this chart in Figure 2.

In prior work, McNutt et al. [46] developed a linting ladder, describing a collection of four properties that broadly describe the behaviors expectable from a linter. This model is the only model of linting usage as far as we are aware. **Checkable** observes that a linter can evaluate whether or not a target has run afoul of the considered property. **Customizable** is the ability to selectively deactivate lints, for instance by noting that a property does not matter in a given context. **Blamable** is the affordance of specificity, highlighting the origin of the issue. Last is **Fixable** or the propensity for automated repair of an identified issue. For instance, consider the code linter ESLint's reaction to a missing semicolon. It can identify that something is missing (checkable), it can be told to ignore that feedback because the style of the current project is to not use semicolons (customizable), where that error occurs (blamable), or even propose a fix like adding a semicolon (fixable). There are other conceivable rungs in this ladder, such as counterfactuals, but these inscribe the behavior of most linters. With this model in hand we now explore a number of our identified linters from Figure 2.

The upper quadrant of Figure 2 is generally well explored by prior work [46], and so will spend most of our time with the more boundary-pushing parts of this diagram. But, briefly, we review the familiar parts as well. Documentation linters (such as HateDetector [65]) operate in the same computational manner (over text) as traditional linters do, but abstract away from them by considering text. McNutt et al.'s [44] statistically mediated metaphor breaks from the purist evaluation perspective by allowing analysis to include execution. FarSight [59] intersects these perspectives to provide an evaluation (and therein warnings) of potential harm from particular prompts by recalling relevant news articles via a model. Color Buddy [46] offers a (input) rebellious perspective on evaluating color palettes by using analysis of colors (which have a

	Evaluation Purist By static analysis	**Evaluation Neutral** By a computer	**Evaluation Rebel** By whatever
Input Purist Input is code	ESLint, gcc, VizLinter [10]	Mirages Linter [44]	StackOverflow downvotes
Input Neutral Input is text	HateDetector [65], word good [19], spell check	Farsight [59]	Community Notes (Sec. 4)
Input Rebel Input is whatever	ColorBuddy [46]	Prof. in Art Sqool [52]	Architecture Planning Review

Figure 2: Different systems organized in an alignment chart format [34]. This framing forgoes nuances between categories—such as human-machine teaming which might exist between evaluation neutral and rebel—and leaves consideration of those interactions to future work.

text representation, but are something different than text) to provide feedback on potential issues that palettes might have. Next we consider the new nonuplets in depth.

3.0.1 Evaluation Neutral / Input Rebel. In the video game Art-Sqool [52], you play a student learning to practice art, particularly in the medium of two-dimensional art. Your efforts are periodically judged by a inscrutable (and satirical) professor whose criteria are opaque. If the criteria are not met your submission is rejected. Viewing the game not as linear ramp, but instead casual creativity instrument [11], we can see this professor as a linter—albeit a suboptimal one. It provides checks if your submission is valid, its feedback can not be customized, it does not offer reasons (outside of some high-level metrics) for its complaints, and does not offer a means for fixing it[1]. While still being run on a computer (and therefore evaluation neutral by Figure 2's framing) this casts the Art Sqool professor not as unassailable judge, but fallible arbiter—analogous to Clippy [62]. Breaking away from objects with text representations that are evaluated through unknown processes allows us to begin seeing linting as something much broader than its traditional spell checker-like functionality.

3.0.2 Evaluation Rebel / Input Purist. Closely aligned with traditional code analysis is the use of downvotes on StackOverflow to indicate low-quality questions or responses. Typically Stack Overflow content consists of a mixture of code and text. While traditional code linters (or documentation linters for that matter) might be applied to these posts, such evaluations can not cover their synthesis, or perhaps, more importantly, their context—e.g. is an answer good if it is a duplicate of three other ones? Fortunately, the other users of StackOverflow can offer a form of human-powered evaluation via downvotes. This type of response is checkable—it runs and can be seen—it has a loose notion of blame—as the down vote is attached to a specific post rather than the whole page (although the specific issue within a post or comment has to be called out textually rather than through UI elements). This form of feedback does not offer automated fixes (although sometimes follow-up comments will include specific ideas) but can be customizable through dialog with the other commenters (perhaps offering a justification on why your identical answer is better than the others). This example highlights the connection between linters and moderation: a policy is being applied, but the context in which it is being used is important.

3.0.3 Evaluation Rebel / Input Rebel. Planning review boards for architectural plans offer a similar human-powered execution of a predefined standard. Within many municipalities there exists a building code: a collection of laws saying what can and can not be built. For instance covering the maximum number of stories a building can have, how much clearance is needed for hallways, the number of required elevators for a given space, and so on. Typically, a developer or architect is required to take their plans for a new building to a review board which considers whether or not the plans are in adherence with the building code (checkable) and highlights which parts are out of compliance (blamable). Interestingly, this type of feedback is customizable, as the planning boards will sometimes offer variances for breaking particular laws—for instance allowing a new tower to exceed the building envelope (sometimes in exchange for a proposed civic good, such as donating funds to cap a noisy highway). This human-powered approach to compliance does suggest that automation may be possible (such as via linter for AutoCAD), but the complexity of applying and navigating these building codes is an impediment necessitating human computation.

3.0.4 Example cases. Before we get to the main topic, let's consider if a couple of everyday things are linters.

First, are the police a form of linter? Clearly they can evaluate behavior (such as speeding) and can provide specific blamable explanations on why they issue a ticket (such as going 100 MPH in a school zone). They can not be customized (unless you view activities like giving that arresting officer a hefty bribe to have not seen you in that school zone as a customization) and they can not provide automated fixes (they only judge past behavior which can not be altered (excluding the likes of 1994's Timecop [28])). More broadly, unlike algorithms, street-level bureaucrats (e.g. cops) refine their decision-making before making a decision [1]. Linters are precisely the opposite of this inconsistently applied, uncorrectable feedback. We suggest then that the police are not linters.

Another example: at the top of many papers on arxiv (such as on the preprint for Lee-Robbins and McNutt [38]) is a badge that warns, "*Important: e-prints posted on arXiv are not peer-reviewed by arXiv; they should not be relied upon without context to guide clinical practice or health-related behavior and should not be reported in news media as established information without consulting multiple experts*

[1] The specific framing within the game is as *art critique*, rather than evaluation, which carries with it a complex set of assumptions (which include viewing critique as a facet of the art-making process rather than an end goal). The connection between critique and linting is a natural one, however we do not explore this metaphor in this work, although future work might do so usefully.

in the field." Is this a lint? It certainly is a check applied, and it can be easily ignored (for instance as it is irrelevant to the aforementioned paper [38]), although there is no remedy (automated or otherwise) or blame (what part of this paper yielded this warning?). It is clear that all lints are warnings of some kind: a failing lint indicates that a policy, law, or standard *may* be broken; not a declaration of a failure but an indication that something may be wrong. But are all warnings lints? Again, we return to the idea that lints offer feedback on something that can be modified rather than only past behavior. So for instance, from an author's perspective, there is nothing we can do that article to get arxiv to remove that badge (short of taking it down, which is not the expressed outcome of the warning badge). From a reader's perspective, it might be taken as a lint to their reading of the work, much in the same way that we consider for Community Notes.

Finally, is Clippy a linter? Clippy was a widely unpopular anthropomorphized paper clip that tried to assist in the use of Microsoft Docs. In his retrospective on why Clippy broadly failed, Whitworth [62] argued that Clippy sealed his fate by being impolite. He did not respect your agency, did not remember or honor your choices, did not offer useful feedback, and so on. We see linters as being a form of polite technology: your choices to ignore them are maintained (e.g. consider the "add to dictionary" button in spell checkers), they offer commentary salient to the domain (and not another), and moreover your agency is centered compared to theirs. There is, of course, an automation bias to consider (it is not polite to be domineering [42]). While there has been some work in this direction [12, 24], developing a sociotechnical theory of respect that encompasses those factors is out of the scope of this work.

4 Exploration: Community Notes as Linters

Now we come to the main topic of this work, Community Notes as linters. While community linters are conceivably usable as a linter for any domain—such as truth in general—we focus on Community Notes evaluating data visualizations. In this section, we'll briefly describe what a Community Note (formerly Birdwatch[2]) is and how it is materialized on a social media platform. Then, we discuss how Community Notes operate as linters on data visualizations. In Figure 3 we provide three examples of Community Notes that we will argue act as linters on data visualizations.

4.1 Anatomy of a Community Note

Here, we provide a brief overview of Community Notes that as relevant to our inquiry. The idea of Community Notes can exist on any social platform (and may become more prevalent in the future), but, at the time of this writing, the X platform has the most widely used and visible implementation. As such, we rely on their particular approach in this paper. A more nuanced description of Community Notes and the details of their creation, approval, and distribution can be found online[3]. Importantly, Community Notes, in their present form, only seek to add contextual information. They do not actively moderate content, for example, by removing or modifying it—this is something only that the original content creator can do.

4.1.1 Generation & Distribution. Any individual can request a Community Note be generated for some content, but at the moment, only a verified collection of individuals (Community Notes contributors) can write notes. Community Notes are written only if there are sufficient requests for a community note to appear next to some content (e.g. one person flagging problematic content is not enough to trigger notage). When a note is written it is then rated by community members as "helpful" (Options: Yes, Somewhat) or "not helpful". Identifying truly helpful and unhelpful Community Notes is not just a matter of aggregating votes, applying a threshold, and filtering for helpful notes. Instead, the data gathered from all contributors and generated notes are modeled to produce a final score of helpfulness.

We do not dive into the modeling details, but it is an interesting read on how to manage sparse data while also taking into account individual preferences and variances from community contributions, without explicitly asking for them.

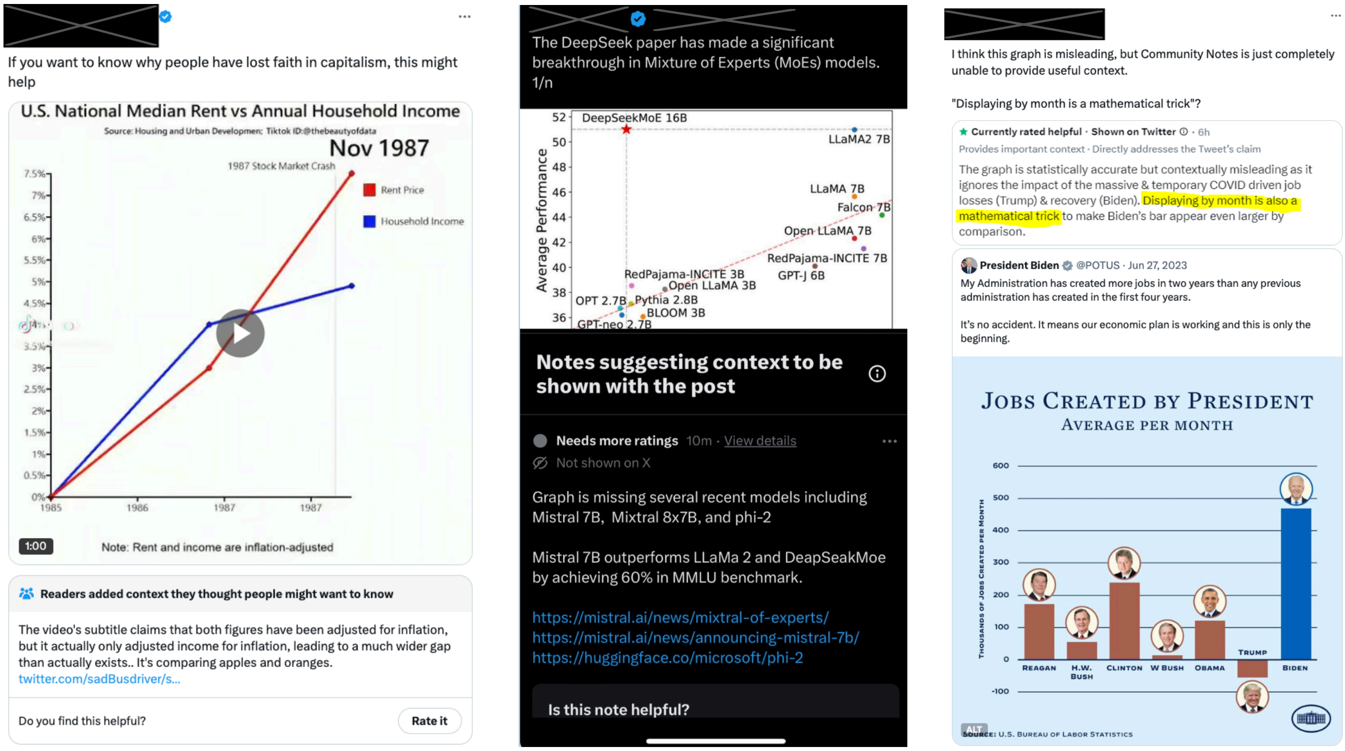
Whether a helpful Community Note is displayed is still dependent on a variety of factors beyond simple 'majority rules'. These factors can be different across platforms. In the current X implementation, Community Notes are also evaluated based on a user's prior rating of notes. The idea is that individuals whose prior ratings are highly correlated likely have similar perspectives compared to individuals with more divergent ratings. The recency of content also plays an important role in determining the utility of Community Notes. All of these factors produce a ranking that is ultimately employed to determine whether to display a Community Note along with some content. When a Community Note is displayed it is featured prominently with the original posted content (as in Figure 3).

4.1.2 Comparison to other Social Media Features. Community notes are not the only means that people can use to provide feedback on some content (or data visualizations). It is possible to post a response to the original content in two ways. 'Replies' can provide the critique in a thread that follows from (and links to) the original content. Alternatively 'Quote Posts' will embed the original content with a person having the option of posing a critique. Quote Posts can act similarly to a Community Note, in that the critique is placed prominently alongside the content, although it lacks the stamp of officiality that Notes have. Both replies and quotes may contain information that could constitute linting, but, it is not clear to what extent this is considered an effective example of human computation.

Like Community Notes, 'replies' and 'quotes' can be posed by anyone, but, unlike Community Notes, they are not directly vetted by a broad group. Indirect methods could substitute the 'wisdom of the crowd', for example, by 'liking' or 'reposting' content, which may cause a critique to gain traction. Thus, in some respect, 'replies', and perhaps more pertinently 'quotes', can also function as a way to leverage human computation for linting activities. But because much of this is indirect as a side-effect of algorithmic content curation, rather than a direct and more focused community effort, we do not consider them further.

[2] https://blog.x.com/en_us/topics/product/2021/introducing-birdwatch-a-community-based-approach-to-misinformation

[3] https://communitynotes.x.com/guide/en/about/introduction



[redacted]

If you want to know why people have lost faith in capitalism, this might help

U.S. National Median Rent vs Annual Household Income
Source: Housing and Urban Developmen; Tiktok ID:@thebeautyofdata
Nov 1987
1987 Stock Market Crash
Rent Price
Household Income
7.5%
7%
6.5%
6%
5.5%
5%
4.5%
4%
3.5%
3%
2.5%
2%
1.5%
1%
0.5%
0%
1985
1986
1987
1987
1:00
Note: Rent and income are inflation-adjusted

Readers added context they thought people might want to know

The video's subtitle claims that both figures have been adjusted for inflation, but it actually only adjusted income for inflation, leading to a much wider gap than actually exists.. It's comparing apples and oranges. twitter.com/sadBusdriver/s...

Do you find this helpful? Rate it

[redacted]

The DeepSeek paper has made a significant breakthrough in Mixture of Experts (MoEs) models. 1/n

Notes suggesting context to be shown with the post

Needs more ratings 10m · View details

Not shown on X

Graph is missing several recent models including Mistral 7B, Mixtral 8x7B, and phi-2

Mistral 7B outperforms LLaMa 2 and DeapSeakMoe by achieving 60% in MMLU benchmark.

https://mistral.ai/news/mixtral-of-experts/
https://mistral.ai/news/announcing-mistral-7b/
https://huggingface.co/microsoft/phi-2

Is this note helpful?

[redacted]

I think this graph is misleading, but Community Notes is just completely unable to provide useful context.

"Displaying by month is a mathematical trick"?

Currently rated helpful · Shown on Twitter · 6h

Provides important context · Directly addresses the Tweet's claim

The graph is statistically accurate but contextually misleading as it ignores the impact of the massive & temporary COVID driven job losses (Trump) & recovery (Biden). Displaying by month is also a mathematical trick to make Biden's bar appear even larger by comparison.

President Biden @POTUS · Jun 27, 2023

My Administration has created more jobs in two years than any previous administration has created in the first four years.

It's no accident. It means our economic plan is working and this is only the beginning.

Jobs Created by President
Average per month
Thousands of jobs created per month
600
500
400
300
200
100
0
-100
Reagan
H.W. Bush
Clinton
W Bush
Obama
Trump
Biden
ALT
Source: U.S. Bureau of Labor Statistics

Figure 3: Example of three Community Notes as applied to data visualizations.

4.2 Human Computation and Linting

Having established that Community Notes are indeed a platform for human computation, by directly recruiting from a broad community and achieving a consensus result (the content and presence of the note), we now return to a central question of this paper: *are Community Notes linters?*

4.2.1 *Examples of Sociotechnical Lints.*

The three Community Notes in Figure 3 broadly present what we would consider to be three examples of lints. 1) In the US National Median Rent vs Annual Household Income animated data visualization, the Community Notes indicate that the two lines have had different statistical treatments. The red line, representing rent price has *not* been adjusted for inflation while that household income *has*, thus exaggerating the differences. The magnitude of the difference is essentially a mirage that results from manipulated scales [44]. 2) Shows the performance of different large language models. The Community Note identifies several missing models that, at the time, outperform some of the top models presented in this visualization (an example of Cherry Picking). 3) The data visualization shows the rate of job growth, per month under different administrations. The Community Note indicates that this choice of aggregate measure is potentially misleading and does not account for other contextual factors (e.g. the volatility of the pandemic). What is perhaps equally interesting in this example is the use of other social features (a quote response) to critique the ambiguity of the Community Note.

4.2.2 *But is it linting?*

Each of the examples shown Figure 3 points to legitimate issues with the data visualizations that, if valid (more on this in a moment), provide important information for the creators and readers of these charts. These examples concern scale, missing data, and choice of aggregation have been explored in prior research for linting visualizations [10, 44] and machine learning datasets [29]. However, in both instances, purely automated strategies (e.g. metamorphic testing, a deep neural network, static analysis) are proposed. Here, the same kinds of evaluations (or lints!) emerge organically in Community Notes.

However, we can go further. In Figure 2 we categorize Community Notes as fitting in the Input Neutral and Evaluation Rebel alignment as their input is (mostly) text and their means of evaluation uses humans (organized by the community notes algorithm). We can also now show that Community Notes conform to the expected behaviors of linters, as outlined in the linting ladder:

Community Notes are Checkable. Notes check against social norms in general, but specifically against misinformation. Interestingly, Community Notes provide no prescriptive definition of *what* misinformation is. The decision is made by the community (echoing our notion of linters as community standards). For data visualizations, it is entirely possible for people to also provide lints on adherence to visualization design principles or guidelines, among other previously established forms of visualization linters [10, 47] as well. But these are almost secondary considerations to the larger question of whether the visualization is misleading.

Community Notes are Customizable. The presence of the note next to some social media content is dependent on the community evaluation, which is in turn established by several criteria. While (currently) individuals using a social media platform do not have the option to selectively deactivate lints, the community does. For example, any individual is free to propose a Community Note, but if there is not enough broad support, the note never materializes – in other words, the lint is deactivated. The ranking algorithm can also be adjusted to further customize whether a Community Note is shown or not. It may be an interesting thought experiment to consider the implications of deactivating social links at the level of individual users (such as via an ad block-style browser extension). Can (should?) a user request to ignore a Community Note in their timeline? Or will this merely filter into a linty bubble?

Community Notes are Blamable. Community notes can provide specific examples of the issue switch a data visualization, as is shown clearly in Figure 3. While the relative precision of these lints is not specified by the format, these can go into details specifics, or highlight generality.

Community Notes can Support Fixable Changes. In evaluating this rung, we debated to what extent Community Notes provide fixable suggestions. One mechanism for fixing misleading content is through social shaming, whereby the original user will actually delete the original content that a Community Note identified as misleading. However, this represents an extreme change – a standard linter would not suggest you should delete your entire code, but suggest more nuance refinements. Editing capabilities are inconsistent on social platforms, but, when they exist, they could provide a more nuanced mechanism to fix original content.

It would seem then that we can see community notes as linters because they do lint-like things (à la being a good person because you do good things, rather than doing good things because you are good person). More broadly they exist with a comparable social contract, as they embodying a dynamic framework for social accountability and refinement. They are checkable, aligning with social norms to identify misinformation, customizable, adapting to collective evaluations, blamable, by pointing out specific shortcomings, and and can even guide fixable improvements. As seen here, community-driven processes can be used to enhance content quality—whether for social media or visualizations or whatever ever—by balancing the rigor of shared standards with the flexibility needed for iterative progress. This interplay of community-driven critique and iterative improvement underscores the broader potential for collaborative systems to foster trust and transparency in diverse domains.

4.3 Beyond Traditional Linters

If we accept that Community Notes are indeed a form of linting powered by human computation, then, they also present some interesting implications beyond what is considered with traditional linters.

A prominent potential bias in linters is the assumption that if the linter passes, there are no problems with the object of analysis. Community-supported linters exhibit an interesting variation on this: if one post sports a Community Note, does that imply that all other posts that do not have Community Notes are fine? Does this broadly imply that everything else in the universe has been deemed acceptable? While this seems absurd, this is more specifically known as the implied truth effect [51]. In the Community Notes algorithm, the factors relating to whether a note appears or not are related to contextual factors (e.g. recency, engagement) that fail to prioritize misleading content even with the same type of lint (e.g. visualizations with scale issues are not consistently flagged). Those present an interesting opportunity to explore a hybrid model, where, once humans identify relevant lints an automated system backfills the rest. Such a mechanism presents its own challenges but is an interesting avenue to explore.

Another interesting difference is how linters respond to unresolveable issues. Something we also observed within our work on linting color palettes (as well as navigating some troublesome ESLint configs), is that in some cases collections of lints are not always resolvable. This can cause the user to fix one issue, and then, in attempting to fix another issue, break the first. An example of an attempt to resolve this issue is VizLinter [10], which tries to propose fixes that fix not just one linted issue by all subsequent ones that arise from fixing that initial issue—however this does lead to strong automation bias. This type of conflicting domain knowledge appears in the context of a Community Note as well. For instance, one could view mutually conflicting communal opinions on a given post as being an example of this. The current approach taken in the Community Notes algorithm is to look for consensus among divergent community bases (e.g. people with different political affiliations both agree that some content is problematic). However, this is an interesting social choice and one that hides the complexity and implications of what may be unresolvable lints. For example, there may be two sizable camps, the first believes pie charts should never be used and another that believes that they are a broadly approachable graphical form. Such conflicts and differing community standards are technically modeled but not presented with a Community Note, and is interesting future work.

Finally there is a question of *who* is doing the linting, which raises questions both of expertise as well as adversarial behavior. In traditional linters the collection of lints are typically prepared by someone expert in that domain. For instance, Lei et al. [39] describe a protracted engagement with a geographer to design their lints for maps. By their nature, community notes draw on individuals who are not necessarily expert in all areas. Kangur et al. [32] describe how external sources are used as validation (to various effects). This situation is more complicated within visualization, which lacks unambiguous means of evaluating visualization quality (either automatic or theoretical). While guidelines exist [15] their validity is often debated [13]. Similarly, the potential for misinformation inherent to this setting creates opportunities for antagonistic actors to create misinformation through the notes themselves. The communal aspects of community notes attempt to circumvent this issue by drawing on the amiability of the crowd. Potentially these issues suggest that something akin to means of external sourcing for visualization analysis are necessary to form useful commentaries.

5 Discussion

This work poses a provocation: what happens if we understand the granular, specific, and actionable feedback typically enshrouded in linters as a computational tool, as something that can be wielded by the crowd, as a representation of their beliefs and values? That is, one of the goals of this paper (particularly as an alt.chi paper) is to explore the effect of recontextualizing linters into a social computing area. In doing so we sought to explore this metaphor in closer depth and more broadly understand its bounds.

Generally we find this metaphorical extension to be informative: it usefully reveals some new things about linters (such as the requirement that they operate over mutable things) and highlights the similarity with tools that we find in more everyday circumstances. Following Carpenter [8], we hope that in posing this thought experiment we prompt the reader to see the world in a new light: what things existing in ambient day-to-day life are linters? Of those that are, what can linters learn from them, and what can they learn from linters (such as fitting onto the rungs of the linting ladder)? Of those that aren't, would making them more lint like be useful? We close this paper with several considerations.

5.0.1 Mandatory AI subsubsection. A recent line of work has explored replicating algorithm's using crowd-work or distributed human computation through LLMs. Wu et al. [69] develop a variety of LLM-based pipelines for replicating crowd-work algorithms. Grunde-McLaughlin et al. [21] develop a broader theory of how these types of pipes parallel one another. In parallel, AI-ification of linters is something that naturally falls out of this comparison. Conceivably, such a combination could usefully operationalize domain knowledge without the expense of human attention. Moreover, it is not without precedence: the original work on AI Chains [68] explored a critique system for visualization that is viewable as a linter. Wang et al. [58] employ a collection of lint-like critics to guide the automated creation of visualization narratives. Human-powered linters seem best suited to problems that are generally difficult to automate or scale, such as moderation [20]. In embracing new technologies, such as AI, to enlarge the remit of linters, we should be careful to consider if such methods are being used to actually expand linters' grasp or merely simplify their implementation.

5.0.2 Linters and uncertainty. As with any human driven computation, there will naturally be some variance compared to the diligent application of programmatic logic found in computer-driven computation. While the algorithmic implementation of tools like Community Notes reduce this variance, this may also give way to implied truth effect biases. While socially mediated tools like Community Notes communicate this uncertainty through a connection with a human element (people are only human and make mistakes), other areas in which human computation might be applied as linter might not do so as clearly. It seems important then, in developing future human-powered linters to consider means of communicating this uncertainty and not let faux-automation (or just dogmatic presentation) lead to unintended ends. Beyond the human linters, communicating uncertainty in lints is a useful future research for lint presentation (although Hopkins et al. [26] explore a related visual metaphor with sketchy rendering) as some tools may give back variable responses—such as in using statistically motivated evaluation methods, as McNutt et al. [44] do, or in using non-deterministic AI-based methods.

5.0.3 Linting Philosophy. Just as all scientific works have a notion of truth that motivates their work, so too do assistant-style tools carry epistemologies of different sorts. For instance, recommendation systems position themselves in such a way a that suggests that there is a correct answer among a possibility space. In many ways, linters are the dual of recommendation systems: rather than suggesting they evaluate, they approach correctness in a manner which is undogmatic and can be overridden following the user's desires (cf. Figure 1). Offenhuber [50] describes a space of visualizations that form themselves—for instance desire footpaths being a (autographic) visualization of people's preference for cutting corners and so forth. Linter's evaluative perspective is hard to reconcile with this naturally occurring process perspective, and so additional consideration of the epistemological underpinnings of linters is needed in order to form comparable evaluations. This may not be generally reconcilable, as linters assume that "incorrect" designs may be fixed and adjusted, which is not necessarily the case for such systems (what would an incorrect desire path mean?).

5.0.4 Limitations and future work. This work offers a reflection on a specific form of linters—from a visualization rather than social computing perspective. While this serves our goal of offering a provocation, it does not explore how general users understand content presented in that way. While there are quite a number of studies about Community Notes (e.g. [16, 32]) and their relationship with trust, ours is the first to investigate this alternative viewpoint. Understanding the role of trust in visualization [18] is a pressing issue, and further consideration of the connection between trust, human-powered moderation-like tools, and visualization seems like useful future research. One means of increasing the trust in visualizations presented on social media might be to create a new social-media platform in which only visualization researchers are allowed to post (perhaps VizSky, VIStter, or V). This will ensure that the graphics presented are of high quality, but in a manner that may be a little boring.

Acknowledgments

We are grateful for the pointers provided by Maxim Lisnic. We appreciate the interesting commentary from our many reviewers.

References

[1] Ali Alkhatib and Michael Bernstein. 2019. Street-Level Algorithms: A Theory at the Gaps Between Policy and Decisions. In *SIGCHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–13. doi:10.1145/3290605.3300760

[2] Mohammad Allahbakhsh, Boualem Benatallah, Aleksandar Ignjatovic, Hamid Reza Motahari-Nezhad, Elisa Bertino, and Schahram Dustdar. 2013. Quality Control in Crowdsourcing Systems: Issues and Directions. *IEEE Internet Computing* 17, 2 (2013), 76–81. doi:10.1109/MIC.2013.20

[3] AntV. 2023. Advisor.lint. ava.antv.antgroup.com/en/api/advice/advisor-lint.

[4] Daniel W Barowy, Emery D Berger, and Benjamin Zorn. 2018. ExceLint: automatically finding spreadsheet formula errors. *Proceedings of the ACM on Programming Languages* 2, OOPSLA (2018), 1–26. doi:10.1145/3276518

[5] Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. In *ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 313–322. doi:10.1145/1866029.1866078

[6] R. Borgo, L. Micallef, B. Bach, F. McGee, and B. Lee. 2018. Information Visualization Evaluation Using Crowdsourcing. *Computer Graphics Forum* 37, 3 (2018), 573–595. doi:10.1111/cgf.13444

[7] David Borland and Russell M Taylor Ii. 2007. Rainbow color map (still) considered harmful. *IEEE computer graphics and applications* 27, 2 (2007), 14–17. doi:10.1109/MCG.2007.46

[8] John Carpenter. 1988. They Live. Universal Pictures.

[9] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative Crowdsourcing for Labeling Machine Learning Datasets. In *SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2334–2346. doi:10.1145/3025453.3026044

[10] Qing Chen, Fuling Sun, Xinyue Xu, Zui Chen, Jiazhe Wang, and Nan Cao. 2021. Vizlinter: A linter and fixer framework for data visualization. *Transactions on Visualization and Computer Graphics* 28, 1 (2021), 206–216. doi:10.1109/TVCG.2021.3114804

[11] Katherine Compton. 2019. *Casual creators: Defining a genre of autotelic creativity support systems*. University of California, Santa Cruz. https://www.escholarship.org/uc/item/4kg8g9gd

[12] Michael Correll. 2021. The Clippy-ization of Human-Computer Design. *Medium* (2021). https://mcorrell.medium.com/the-clippy-ization-of-human-computer-design-c66e8042de88

[13] Michael Correll and Michael Gleicher. 2014. Bad for Data, Good for the Brain : Knowledge-First Axioms For Visualization Design. In *DECISIVe : Workshop on Dealing with Cognitive Biases in Visualisations*. http://graphics.cs.wisc.edu/Papers/2014/CG14a

[14] Alan Cruickshank. 2023. Rolling out SQLFluff with a new team. docs.sqlfluff.com/en/stable/teamrollout.html.

[15] Alexandra Diehl, Alfie Abdul-Rahman, Mennatallah El-Assady, Benjamin Bach, Daniel A. Keim, and Min Chen. 2018. VisGuides: A Forum for Discussing Visualization Guidelines. In *Eurographics Conference on Visualization*. Eurographics Association, 61–65. doi:10.2312/EUROVISSHORT.20181079

[16] Chiara Patricia Drolsbach, Kirill Solovev, and Nicolas Pröllochs. 2024. Community notes increase trust in fact-checking on social media. *PNAS Nexus* (2024), pgae217. doi:10.1093/pnasnexus/pgae217

[17] Eleventy. 2024. Inclusive Language Plugin. https://www.11ty.dev/docs/plugins/inclusive-language/.

[18] Hamza Elhamdadi, Adam Stefkovics, Johanna Beyer, Eric Moerth, Hanspeter Pfister, Cindy Xiong Bearfield, and Carolina Nobre. 2023. Vistrust: a multidimensional framework and empirical study of trust in data visualizations. *IEEE Transactions on Visualization and Computer Graphics* (2023). doi:10.1109/TVCG.2023.3326579

[19] Brian Ford. 2019. Write Good: Naive linter for English prose . github.com/btford/write-good.

[20] Robert Gorwa, Reuben Binns, and Christian Katzenbach. 2020. Algorithmic content moderation: Technical and political challenges in the automation of platform governance. *Big Data & Society* 7, 1 (2020), 2053951719897945. doi:10.1177/2053951719897945

[21] Madeleine Grunde-McLaughlin, Michelle S Lam, Ranjay Krishna, Daniel S Weld, and Jeffrey Heer. 2023. Designing LLM chains by adapting techniques from crowdsourcing workflows. *arXiv preprint arXiv:2312.11681* (2023). doi:10.48550/arXiv.2312.11681

[22] Han L Han, Miguel A Renom, Wendy E Mackay, and Michel Beaudouin-Lafon. 2020. Textlets: Supporting constraints and consistency in text documents. In *SIGCHI Conference on Human Factors in Computing Systems*. 1–13. doi:10.1145/3313831.3376804

[23] Kotaro Hara, Vicki Le, and Jon Froehlich. 2013. Combining crowdsourcing and google street view to identify street-level accessibility problems. In *SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 631–640. doi:10.1145/2470654.2470744

[24] Jeffrey Heer. 2019. Agency plus automation: Designing artificial intelligence into interactive systems. *Proceedings of the National Academy of Sciences* 116, 6 (2019), 1844–1850. doi:10.1073/pnas.1807184115

[25] Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 203–212. doi:10.1145/1753326.1753357

[26] Aspen K Hopkins, Michael Correll, and Arvind Satyanarayan. 2020. Visualint: Sketchy in situ annotations of chart construction errors. In *Computer Graphics Forum*. Wiley, 219–228. doi:10.1111/cgf.13975

[27] Matthew Hull, Vivian Pednekar, Hannah Murray, Nimisha Roy, Emmanuel Tung, Susanta Routray, Connor Guerin, Justin Chen, Zijie J. Wang, Seongmin Lee, Mahdi Roozbahani, and Duen Horng Chau. 2023. VISGRADER: Automatic Grading of D3 Visualizations. *Transactions on Visualization and Computer Graphics* (2023). doi:10.1109/TVCG.2023.3327181

[28] Peter Hyams. 1994. TimeCop. Universal Pictures.

[29] Nick Hynes, D Sculley, and Michael Terry. 2017. Data linter: Lightweight, automated sanity checking for ml data sets. In *NIPS MLSys Workshop*.

[30] Steven J Jackson and Sarah Barbrow. 2015. Standards and/as innovation: Protocols, creativity, and interactive systems development in ecology. In *SIGCHI Conference on Human Factors in Computing Systems*. 1769–1778. doi:10.1145/2702123.2702564

[31] Alex Kale, Ziyang Guo, Xiao Li Qiao, Jeffrey Heer, and Jessica Hullman. 2023. EVM: Incorporating model checking into exploratory visual analysis. *Transactions on Visualization and Computer Graphics* (2023). doi:10.1109/TVCG.2023.3326516

[32] Uku Kangur, Roshni Chakraborty, and Rajesh Sharma. 2024. Who Checks the Checkers? Exploring Source Credibility in Twitter's Community Notes. *arXiv preprint arXiv:2406.12444* (2024). doi:10.48550/arXiv.2406.12444

[33] Hieke Keuning, Johan Jeuring, and Bastiaan Heeren. 2023. A Systematic Mapping Study of Code Quality in Education. In *Conference on Innovation and Technology in Computer Science Education V. 1*. 5–11. doi:10.1145/3587102.3588777

[34] Know Your Meme. 2022. Alignment Charts - Salad Alignment Chart. https://knowyourmeme.com/photos/2382569-alignment-charts Online meme database entry.

[35] Robert Kosara. 2024. Business Data Visualization, Beyond the Boring . *IEEE Computer Graphics and Applications* 44, 05 (Sept. 2024), 153–158. doi:10.1109/MCG.2024.3407489

[36] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2017. Chapter 14 - Online and ubiquitous HCI research. In *Research Methods in Human Computer Interaction (Second Edition)* (second edition ed.). Morgan Kaufmann, Boston, 411–453.

[37] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2017. Online and Ubiquitous HCI Research. In *Research Methods in Human Computer Interaction*. Morgan Kaufmann, Chapter 14, 411–453.

[38] Elsie Lee-Robbins and Andrew McNutt. 2023. Only YOU Can Make IEEE VIS Environmentally Sustainable. *alt.vis* (2023). doi:10.48550/arXiv.2308.15429 https://arxiv.org/abs/2308.15429.

[39] Fan Lei, Arlen Fan, Alan M MacEachren, and Ross Maciejewski. 2023. GeoLinter: A Linting Framework for Choropleth Maps. *Transactions on Visualization and Computer Graphics* (2023). doi:10.1109/TVCG.2023.3322372

[40] Maxim Lisnic, Zach Cutler, Marina Kogan, and Alexander Lex. 2025. Visualization Guardrails: Designing Interventions Against Cherry-Picking in Interactive Data Explorers. (2025). To Appear.

[41] Kurt Luther, Jari-Lee Tolentino, Wei Wu, Amy Pavel, Brian P Bailey, Maneesh Agrawala, Björn Hartmann, and Steven P Dow. 2015. Structuring, aggregating, and evaluating crowdsourced design critique. In *ACM Conference on Computer Supported Cooperative Work & Social Computing*. 473–485. doi:10.1145/2675133.2675283

[42] Andrew McNutt, Anamaria Crisan, and Michael Correll. 2020. Divining insights: Visual analytics through cartomancy. In *SIGCHI Conference on Human Factors in Computing Systems (Extended Abstracts)*. 1–16. doi:10.1145/3334480.3381814

[43] Andrew McNutt and Gordon Kindlmann. 2018. Linting for visualization: Towards a practical automated visualization guidance system. In *VisGuides (IEEEVIS Workshop)*. 1–14.

[44] Andrew McNutt, Gordon Kindlmann, and Michael Correll. 2020. Surfacing visualization mirages. In *SIGCHI Conference on Human Factors in Computing Systems*. 1–16. doi:10.1145/3313831.3376420

[45] Andrew McNutt, Anton Outkine, and Ravi Chugh. 2023. A Study of Editor Features in a Creative Coding Classroom. In *SIGCHI Conference on Human Factors in Computing Systems*. 1–15. doi:10.1145/3544548.3580683

[46] Andrew McNutt, Maureen C Stone, and Jeffrey Heer. 2024. Mixing Linters with GUIs: A Color Palette Design Probe. *IEEE Transactions on Visualization and Computer Graphics* (2024). doi:10.1109/TVCG.2024.3456317

[47] Andrew McNutt, Chenglong Wang, Robert A Deline, and Steven M Drucker. 2023. On the design of ai-powered code assistants for notebooks. In *SIGCHI Conference on Human Factors in Computing Systems*. 1–16. doi:10.1145/3544548.3580940

[48] Joseph Bahman Moghadam, Rohan Roy Choudhury, HeZheng Yin, and Armando Fox. 2015. AutoStyle: Toward Coding Style Feedback at Scale. In *ACM Conference on Learning Scale*. Association for Computing Machinery, New York, NY, USA, 261–266. doi:10.1145/2818052.2874315

[49] Arpit Narechania, Adam Coscia, Emily Wall, and Alex Endert. 2021. Lumos: Increasing awareness of analytic behavior during visual data analysis. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2021), 1009–1018. doi:10.1109/TVCG.2021.3114827

[50] Dietmar Offenhuber. 2023. *Autographic Design: The Matter of Data in a Self-Inscribing World*. MIT Press. doi:10.7551/mitpress/14666.003.0010

[51] Gordon Pennycook, Adam Bear, Evan T Collins, and David G Rand. 2020. The implied truth effect: Attaching warnings to a subset of fake news headlines increases perceived accuracy of headlines without warnings. *Management science* 66, 11 (2020), 4944–4957. doi:10.1287/mnsc.2019.3478

[52] Polygon. 2019. Art Sqool took me back to real-life art school. https://www.polygon.com/2019/3/1/18246354/art-sqool-impressions-pc-mac.

[53] Alexander J. Quinn and Benjamin B. Bederson. 2011. Human computation: a survey and taxonomy of a growing field. In *SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1403–1412. doi:10.1145/1978942.1979148

[54] Katharina Reinecke and Krzysztof Z Gajos. 2015. LabintheWild: Conducting large-scale online experiments with uncompensated samples. In *ACM conference on computer supported cooperative work & social computing*. 1364–1378. doi:10.

1145/2675133.2675246

[55] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Honolulu, Hawaii, 254–263. https://aclanthology.org/D08-1027/

[56] Nicole Sultanum, Dennis Bromley, and Michael Correll. 2024. Data Guards: Challenges and Solutions for Fostering Trust in Data. *IEEE Visualization Conference* (2024). doi:10.1109/VIS55277.2024.00019

[57] Twitter Notes [n. d.]. Note ranking algorithm. https://communitynotes.x.com/guide/en/under-the-hood/ranking-notes. doi:10.1007/s10288-003-0021-7 Accessed 10/29/2024.

[58] Huichen Will Wang, Larry Birnbaum, and Vidya Setlur. 2025. Jupybara: Operationalizing a Design Space for Actionable Data Analysis and Storytelling with LLMs. In *SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1–13.

[59] Zijie J Wang, Chinmay Kulkarni, Lauren Wilcox, Michael Terry, and Michael Madaio. 2024. Farsight: Fostering Responsible AI Awareness During AI Application Prototyping. In *SIGCHI Conference on Human Factors in Computing Systems*. doi:10.1145/3613904.364233

[60] Colin Ware and Matthew D Plumlee. 2013. Designing a better weather display. *Information Visualization* 12, 3-4 (2013), 221–239. doi:10.1177/1473871612465214

[61] Colin Ware, Maureen Stone, and Danielle Albers Szafir. 2023. Rainbow colormaps are not all bad. *IEEE Computer Graphics and Applications* 43, 3 (2023), 88–93. doi:10.1109/MCG.2023.3246111

[62] Brian Whitworth. 2005. Polite computing. *Behaviour & Information Technology* 24, 5 (2005), 353–363. doi:10.1080/01449290512331333700

[63] Wikipedia. 2024. Colorless green ideas sleep furiously — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Colorless%20green%20ideas%20sleep%20furiously&oldid=1258719573. [Online; accessed 24-November-2024].

[64] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. 2012. Strategies for crowdsourcing social data analysis. In *SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 227–236. doi:10.1145/2207676.2207709

[65] Hana Winchester, Ebtesam Al Haque, Alicia Boyd, and Brittany Johnson. 2023. HaTe Detector: A Tool for Detecting and Correcting Harmful Terminology in Computing Artifacts. In *Symposium on Visual Languages and Human-Centric Computing*. IEEE, 245–248. doi:10.1109/VL-HCC57772.2023.00043

[66] Jo Wood, Alexander Kachkaev, and Jason Dykes. 2018. Design exposition with literate visualization. *Transactions on Visualization and Computer Graphics* 25, 1 (2018), 759–768. doi:10.1109/TVCG.2018.2864836

[67] Aoyu Wu, Liwenhan Xie, Bongshin Lee, Yun Wang, Weiwei Cui, and Huamin Qu. 2021. Learning to Automate Chart Layout Configurations Using Crowdsourced Paired Comparison. In *SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 14, 13 pages. doi:10.1145/3411764.3445179

[68] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. AI Chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *SIGCHI Conference on Human Factors in Computing Systems*. 1–22. doi:10.1145/3491102.3517582

[69] Tongshuang Wu, Haiyi Zhu, Maya Albayrak, Alexis Axon, Amanda Bertsch, Wenxing Deng, Ziqi Ding, Bill Guo, Sireesh Gururaja, Tzu-Sheng Kuo, et al. 2023. Llms as workers in human-computational algorithms? replicating crowdsourcing pipelines with llms. *arXiv preprint arXiv:2307.10168* (2023). doi:10.48550/arXiv.2307.10168

[70] Anbang Xu, Shih-Wen Huang, and Brian Bailey. 2014. Voyant: generating structured feedback on visual designs using a crowd of non-experts. In *ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. Association for Computing Machinery, New York, NY, USA, 1433–1444. doi:10.1145/2531602.2531604

[71] Nicholas C. Zakas and Nicole Sullivan. 2011. CSS Lint. csslint.net.

[72] Wei Zheng and Youssef Sherif. 2019. Visualization Linter.