

Position Paper: **Validatory Visualization** for Human-GenAI Co-Creation



Yu Zhang¹
presenter



Yuheng Zhao²



Yuhan Guo³



Guozheng Li⁴



Siming Chen²






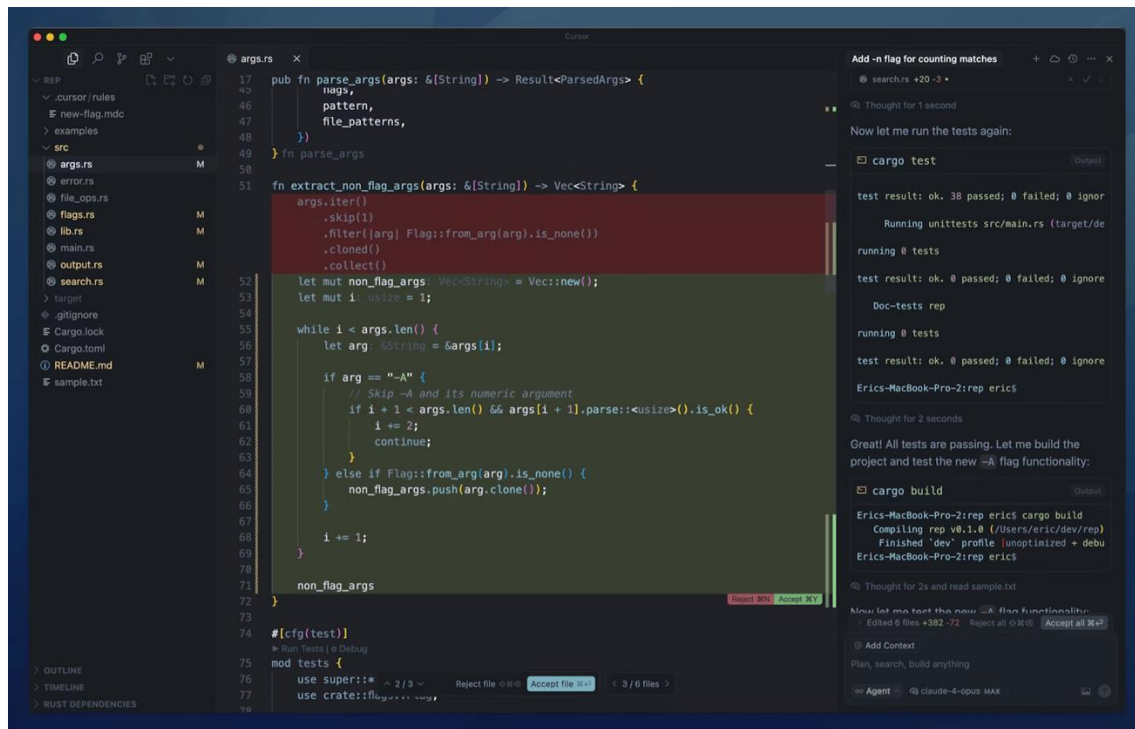
Xiaoru Yuan³

¹University of Oxford ²Fudan University ³Peking University ⁴BIT



Background: Validatory VIS for AI

- AI in content creation apps: code editor, writing assistant, ...
- **Validatory paradigm:**  steer →  generate →  validate
- **Validatory VIS:** (*in-situ, minimalist*) VIS for validating AI output



```
pub fn parse_args(args: &[String]) -> Result<ParsedArgs> {
    flags,
    pattern,
    file_patterns,
}
}
fn parse_args
}

fn extract_non_flag_args(args: &[String]) -> Vec<String> {
    args.iter()
        .skip(1)
        .filter(|arg| Flag::from_arg(arg).is_none())
        .cloned()
        .collect()
}

let mut non_flag_args: Vec<String> = Vec::new();
let mut i: usize = 1;

while i < args.len() {
    let arg = args[i];

    if arg == "-A" {
        // Skip -A and its numeric argument
        if i + 1 < args.len() && args[i + 1].parse::<usize>().is_ok() {
            i += 2;
            continue;
        }
    } else if Flag::from_arg(arg).is_none() {
        non_flag_args.push(arg.clone());
    }

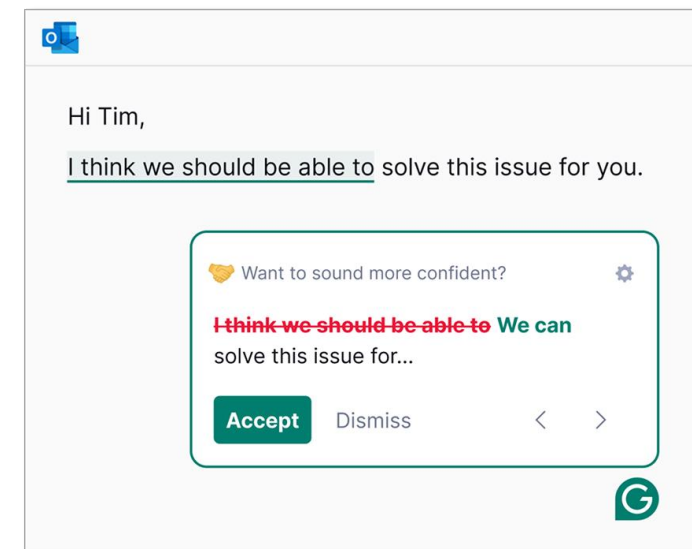
    i += 1;
}

non_flag_args
}

#[cfg(test)]
mod tests {
    use super::*;

    use crate::flag::Flag;
}
```

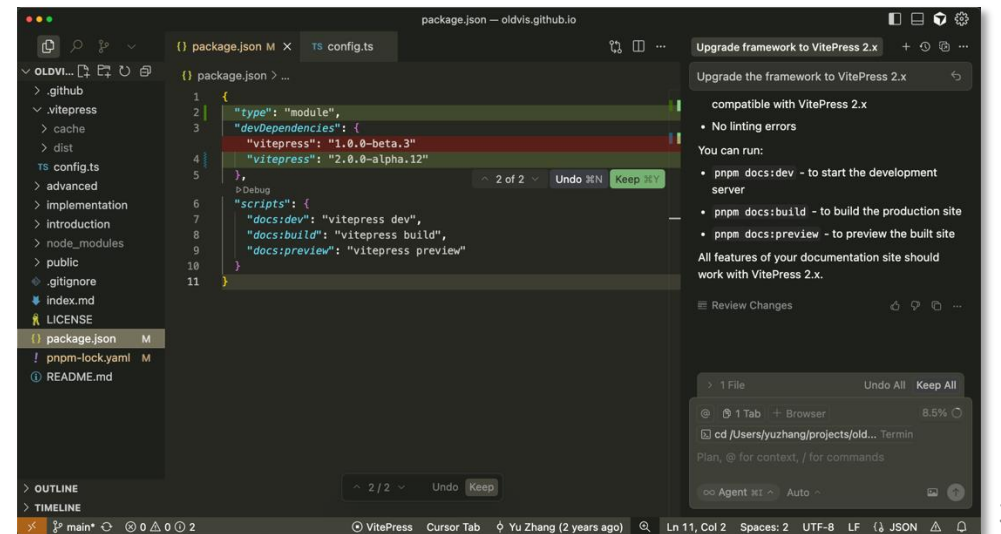
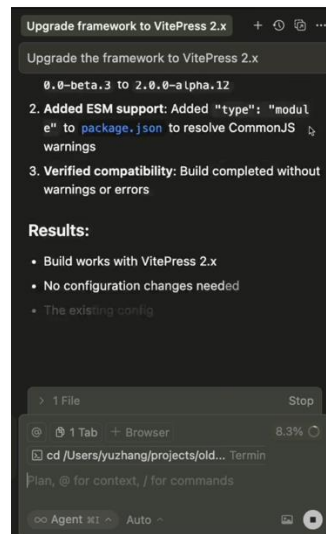
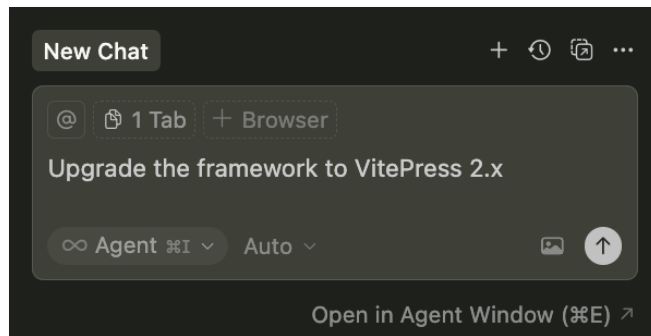
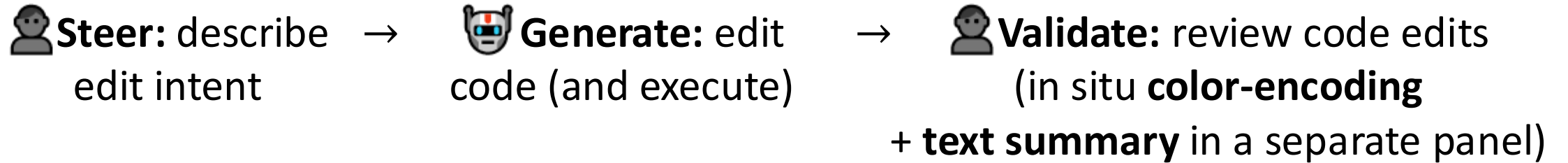
Example – code editor: Cursor







Example – writing assistant: Grammarly

Background: Validatory VIS for AI

- Validatory paradigm: 🧑 steer → 🤖 generate → 🧑 validate
- Validatory VIS: (*in-situ, minimalist*) VIS for validating AI output
- **Example:** 📦 Cursor




Background: Validatory VIS for AI

- Validatory paradigm:  steer →  generate →  validate
- Validatory VIS: (*in-situ, minimalist*) VIS for validating AI output
- **Example:**  Grammarly

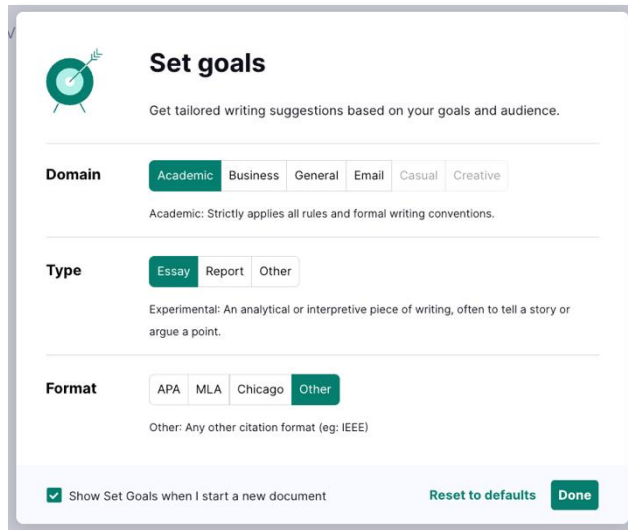
 **Steer:** config writing goals



 **Generate:** check language issues and suggest fixes



 **Validate:** review suggestions (in situ **color-encoding** + **list** in a separate panel)



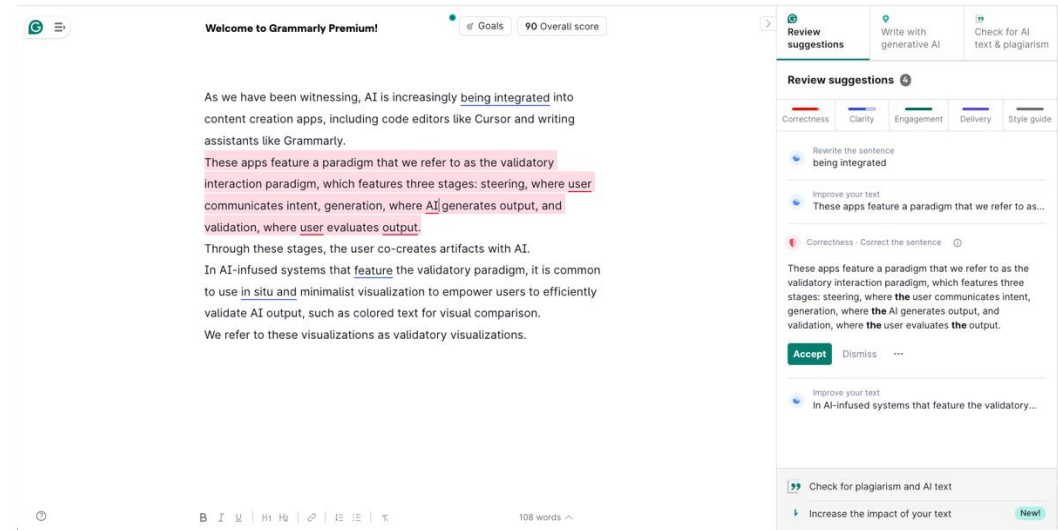
Set goals
Get tailored writing suggestions based on your goals and audience.

Domain
Academic Business General Email Casual Creative
Academic: Strictly applies all rules and formal writing conventions.

Type
Essay Report Other
Experimental: An analytical or interpretive piece of writing, often to tell a story or argue a point.

Format
APA MLA Chicago Other
Other: Any other citation format (eg: IEEE)

Show Set Goals when I start a new document Reset to defaults Done



Welcome to Grammarly Premium! 9 Goals 90 Overall score

As we have been witnessing, AI is increasingly being integrated into content creation apps, including code editors like Cursor and writing assistants like Grammarly.

These apps feature a paradigm that we refer to as the **validatory interaction paradigm**, which features three stages: steering, where user communicates intent, generation, where AI generates output, and validation, where user evaluates output.

Through these stages, the user co-creates artifacts with AI.

In AI-infused systems that feature the **validatory paradigm**, it is common to use **in situ** and **minimalist** visualization to empower users to efficiently validate AI output, such as colored text for visual comparison.

We refer to these visualizations as **validatory visualizations**.

Review suggestions
Write with generative AI | Check for AI text & plagiarism

Review suggestions

Correctness Clarity Engagement Delivery Style guide

Rewrite the sentence
being integrated

Improve your text
These apps feature a paradigm that we refer to as...

Correctness Correct the sentence

These apps feature a paradigm that we refer to as the **validatory interaction paradigm**, which features three stages: steering, where the user communicates intent, generation, where the AI generates output, and validation, where the user evaluates the output.

Accept Dismiss ...

Improve your text
In AI-infused systems that feature the **validatory**...

Check for plagiarism and AI text

Increase the impact of your text New!

108 words

Argument Chain

① GenAI **cannot fully automate** many generative tasks.

① ⇒ ② **Human validation** is usually required in GenAI applications.

③ **Many everyday tasks are generative** (e.g., writing) in nature and benefit from GenAI.

②, ③ ⇒ ④ **Many ordinary users need to validate** AI-generated content.

⑤ GenAI produces content at a **high rate**.

⑥ **Visualization amplifies** human information processing abilities.

④, ⑤, ⑥ ⇒ ⑦ **Validatory visualization is valuable for many ordinary users.**

The Value of Visualization **focus of validatory visualization**

widely studied in VIS

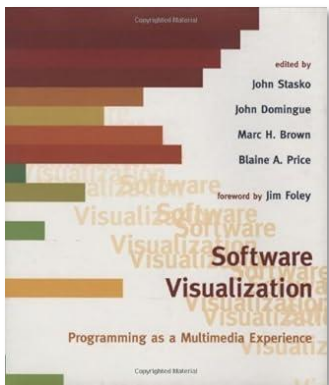
van Wijk, VIS, 2005:

*“a great visualization method is **used by many people**, who use it routinely to **obtain highly valuable knowledge**, **without having to spend time and money** on hardware, software, and effort.”*

Opportunities

Validatory visualization for productivity tools

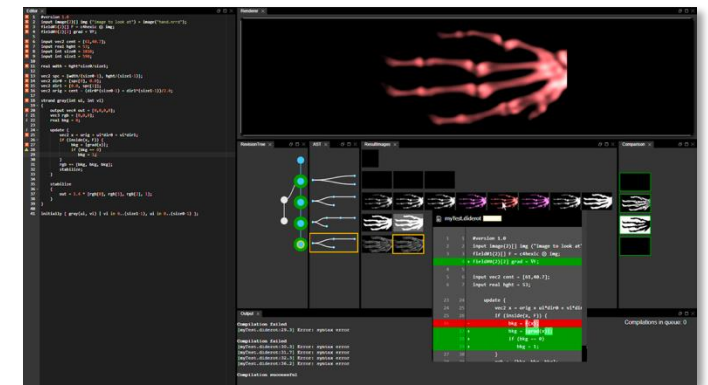
- **Q:** Can we improve Cursor's design for showing AI code edits?
- Relevant VIS topics: **software visualization**, **visual comparison**



Software Visualization,
Stasko et al., 1998

```
151 {
152   "type": "text",
153   "from": {"data": "drive"},
154   "properties": {
155     "enter": {
156       "x": {"scale": "x", "field": "miles"},
157       "y": {"scale": "y", "field": "gas"},
158       "dx": {"scale": "dx", "field": "side"},
159       "dy": {"scale": "dy", "field": "side"},
160       "fill": {"scale": "c", "signal": "mouseover_year"},
161       "text": {"field": "year"},
162       "align": {"scale": "align", "field": "side"},
163       "baseline": {"scale": "base", "field": "side"}
164     }
165   }
166 }
```

Augmenting Code with In Situ Visualizations to Aid Program Understanding, Hoffswell et al., CHI, 2018



Vis-a-Vis: Visual Exploration of Visualization Source Code Evolution, Bolte and Bruckner, TVCG, 2021

Comparative Elements
Targets
Actions (on Relationships)


Comparative Challenges
Number of Targets
Large or Complex Targets
Complex Relationships



Scalability Strategies
Scan Sequentially
Select Subset
Summarize Somehow




Comparative Designs
Juxtapose
Superpose
Explicit Encoding

Fig. 1. Overview of the four considerations of comparison and the abstract categories they impose.

Cost Modeling

Cost of manual paradigm (

Validatory paradigm ( + )

1.  **Steering:** User requests AI to generate/edit content. (cost: C_{steer}^{user})
2.  **Generation:** AI generates content (cost: $C_{generate}^{machine}$)
3.  **Validation:** User validates AI-generated content. (cost: $C_{validate}^{user}$)
 - With $p_{complete}$, the content is acceptable. Task **completed**.
 - With p_{refine} , the user manually refines the content. Task **completed**. (cost: C_{refine}^{user})
 - With $1 - p_{complete} - p_{refine}$, the user **goes back to the steering step**.

Expected task completion cost:

$$C = \sum_{n=1}^{\infty} (1-p)^{n-1} (pnC_{loop} + p_{refine}C_{refine}^{user}) = \boxed{\frac{1}{p}C_{loop} + \frac{1}{p}p_{refine}C_{refine}^{user}}$$

where:






$$C_{loop} = C_{steer}^{user} + C_{generate}^{machine} + C_{validate}^{user} \text{ and } p = p_{complete} + p_{refine}$$

Cost Modeling – Interpretation

Benefit of validatory paradigm: $r = 1 - C/C_{\text{manual}} = 1 - \frac{c_{\text{loop}} + p_{\text{refine}} c_{\text{refine}}^{\text{user}}}{p c_{\text{generate}}^{\text{user}}}$

 When is the validatory paradigm beneficial (i.e., $r > 0$)?


$c_{\text{loop}} + p_{\text{refine}} c_{\text{refine}}^{\text{user}} < p c_{\text{generate}}^{\text{user}}$ when:

- () AI is skilled (large p , small p_{refine})
- () Manual content generation is costly (large $c_{\text{generate}}^{\text{user}}$)
- (  ) Swift steer-generate-validate loop (small c_{loop} and $c_{\text{refine}}^{\text{user}}$)

Cost Modeling – Interpretation

Benefit of validatory paradigm: $r = 1 - \frac{c_{loop} + p_{refine} c_{refine}^{user}}{p c_{generate}^{user}}$

 Compare $c_{validate}^{user}$ and $c_{generate}^{user}$ (informally)

- Evidence 1: P \neq NP belief implies $c_{validate}^{user} < c_{generate}^{user}$ (in certain complexity classes).
- Evidence 2: RLHF assumes rating is simpler than generating ($c_{validate}^{user} < c_{generate}^{user}$).
- Interpretation: **The validatory paradigm converts a complex task to a simpler task.**
 -  Note: AI's competence determines how many times the simpler task needs to be executed.

Issues of the Validatory Paradigm

Assumptions

- Output-centric
- Cost-sensitive

Criticisms

- Automation bias
- (Potential) loss of context, serendipity, creativity, and frictions

Conclusion

- Validatory visualization deserves more attention
- **VIS opportunities:** productivity tools, mixed-initiative data analysis

 **Paper URL:**

<https://zhangyu94.github.io/projects/2025/Zhang2025Validatory/paper.pdf>

Email: yuzhang94@outlook.com

Website: zhangyu94.github.io

Paper QR Code:



 **Related Talks:**

ProactiveVA, LightVA
by Yuheng Zhao

**@Session: Agentic Visualization
and Intelligent Systems**

3:45pm - 5:00pm

Wednesday, Nov 5

