How do Agentic-PRs change code (e.g., additions, deletions, files touched)? How consistent are their descriptions with the actual code changes?

RIAZUL ISLAM RIFAT, BRAC UNIVERSITY

Abstract– AI coding agents are increasingly integrated into software development workflows. Understanding how their code contributions differ and how accurate their descriptions represent the changes they make is crucial. Using the AIDev dataset this study addresses "How do Agentic-PRs change code, and how consistent are their descriptions with actual code changes?" by comprehensive quantitative statistical analysis with qualitative text–code semantic consistency analysis on 33,580 Agentic-PRs from five agents Claude Code, Copilot, Cursor, Devin and Codex. My findings reveal that changes are typically small but vary significantly by agent, with moderate description consistency that declines for larger patches. These findings highlight opportunities for improving human-AI collaboration in Software Engineering (SE) 3.0.

I. INTRODUCTION

The rise of AI coding agents marks a pivotal shift in software engineering where humans and AI collaborate as teammates on core development tasks. These agents, such as GitHub Copilot or Claude Code, now author thousands of pull requests (PRs) daily, automating everything from bug fixes to feature additions. This growing adoption raises questions about the nature of their contributions. How extensive are the code changes they propose? Do their PR descriptions accurately reflect the patches, or do discrepancies risk misleading human reviewers? These questions are crucial for trust, review effort, and software quality. I address this gap by analyzing the AIDev dataset, which contains over 900,000 Agentic-PRs authored by five popular agents (Claude Code, Copilot, Cursor, Devin and Codex) across 116,211 repositories involving 72,189 developers.

Specifically this paper tackles research question 2a from the AIDev preprint "How do Agentic-PRs change code (e.g., additions, deletions, files touched)? How consistent are their descriptions with the actual code changes?" Using AIDev's curated subset (33,580 Agentic-PRs from repositories with more than 100 GitHub stars), I conduct a mixed-methods analysis. Quantitatively, I employ non-parametric statistics to compare the code change metrics across agents. Qualitatively, I use semantic embeddings to measure description-patch alignment.

The results reveal that Agentic-PRs favor small, focused modifications (median: 2 files touched), but agents differ markedly like, GitHub Copilot tends toward minimal changes, while Claude Code handles broader scopes. Description consistency averages 0.414 (cosine similarity), varying by agent (OpenAI Codex highest at 0.489) and weakly correlating with PR size. These insights help to understand the capability and consistency of agentic-PRs across AI coding agents.

II. METHODOLOGY

Data Processing:

To conduct the research the AIDev dataset is used. This dataset comprises 932,791 Agentic PRs authored by five agents: OPENAI CODEX, DEVIN, GITHUB COPILOT, CURSOR, and CLAUDE CODE, across 116,211 repositories involving 72,189 developers (dataset cutoff: August 1, 2025). Each PR is linked to its corresponding repository and developer, along with additional metadata. A curated subset of 33,596 Agentic-PRs from 2,807 repositories with more than 100 GitHub stars is available within the dataset. This enriched subset is used to create a separate merged dataset for the work.

To create the merged dataset – the file level changes (of the pr_commit details subset) are aggregated and grouped by 'pr_id' (Pull Request ID). This aggregation basically represents the PR level changes (per 'pr_id') and contains aggregations like total additions & deletions, total touched files, and total changes for a pull request. Then I merge each PR level change (per 'pr_id') with each PR's agent, title and body (PR wise information is available in the curated pull_request subset). After that I create a new column named full_description in the merged dataset which contains clean text of combined title & body. Finally, this merged dataset (shape: 33580, 10) is used for the analysis

Analysis:

To answer the question "How do Agentic-PRs change code (e.g., additions, deletions, files touched)?" a comprehensive Quantitative Analysis and to answer "How consistent are their descriptions with the actual code changes?" a comprehensive Qualitative Analysis has been done.

Quantitative Analysis: [Scale and Agent Differences]

To assess change scale, I extracted metrics: total_additions, total_deletions, and files_touched per PR. Given non-normal distributions (Figure 1.1), along with general statistical analysis I used the following non-parametric methods:

- Kruskal–Wallis tests to compare distributions across agents.
- Mann-Whitney U tests for pairwise comparison; with Cliff's Delta effect sizes to measure practical significance.
- Visualized the results using several plots.

Qualitative Analysis:

For qualitative analysis, in the merged dataset I created another column named 'code_summary' which contains PR wise files information like file name, status, file's additions and deletions.

I measured alignment between PR bodies (descriptions) and code patches (diffs; code_summary) using SentenceTransformers' all-MiniLM-L6-v2 model which is a compact embedding model fine-tuned for semantic tasks. It encodes texts into vectors and cosine similarity yields scores (-1 to 1). A score towards 1 means that sentences are similar semantically and a score around -1 means sentences contain opposite meanings.

- Computed mean similarity and quartiles.
- Grouped by agent; ANOVA tested differences (statsmodels).
- Pearson correlations linked similarity to quantitative metrics.

This approach scales to large data while capturing nuanced semantics, unlike keyword-based methods.

III. Results [Quantitative Analysis]

To establish a baseline understanding of Agentic-PRs, I first computed descriptive statistics for code additions, deletions, and files touched. Across all agents, most PRs involve very small changes, with a long tail of larger edits. This indicates that while agentic contributions are typically minimal, some agents occasionally generate disproportionately large patches.

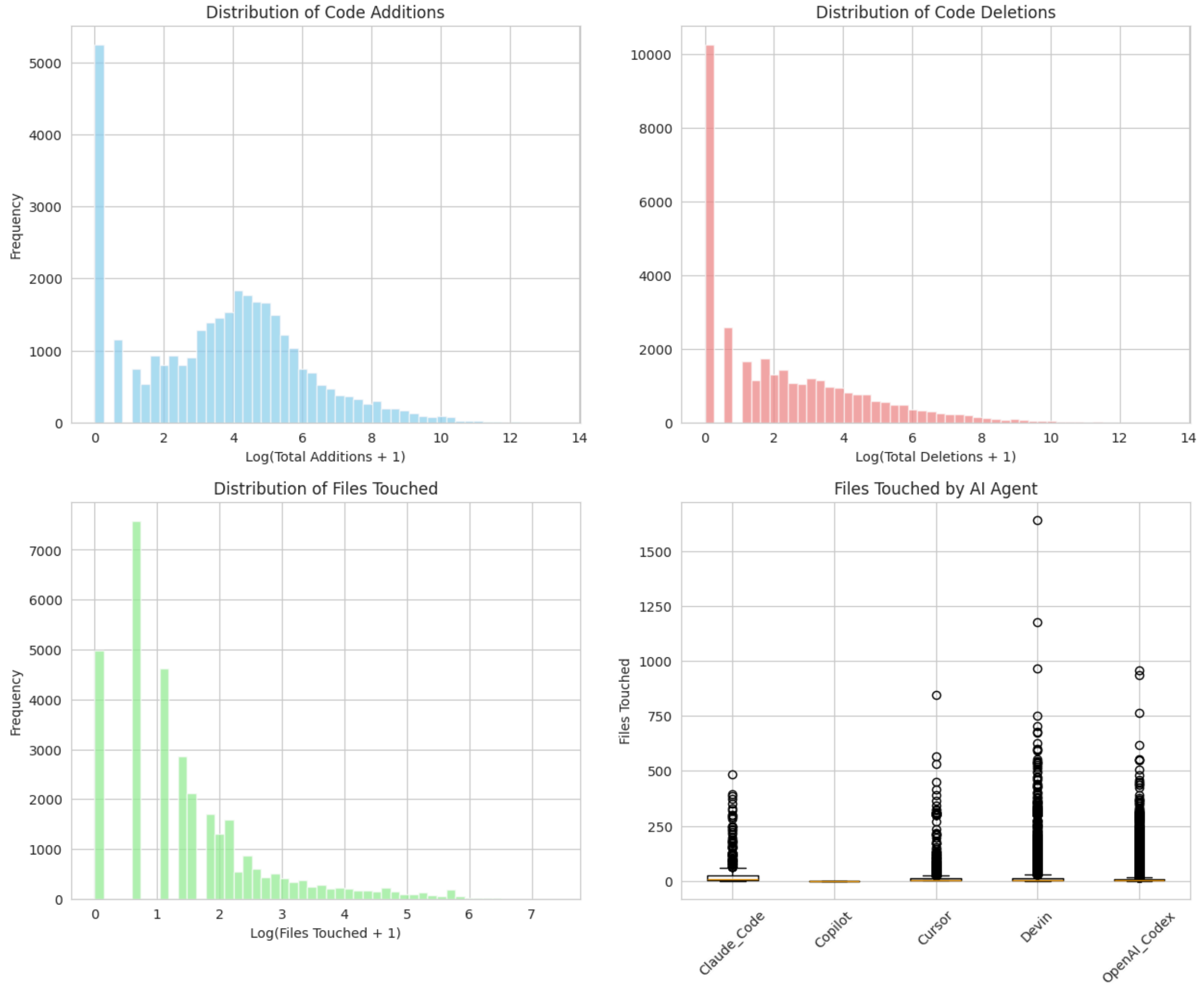


Figure: 1.1

Figure 1.1 illustrates the log-transformed distributions of code additions, deletions, and files touched. The boxplot in Figure 1.1 compares the number of

files touched across agents. Copilot PRs are consistently small in scope, rarely exceeding a handful of files. In contrast, Codex and Devin show a higher variance, with some PRs spanning hundreds (around 500) of files. Cursor and Claude fall between these extremes.
These results confirm that agentic code contributions are usually small but vary significantly across agents.

To test whether differences in code change characteristics across AI agents were statistically significant, I applied the Kruskal–Wallis H test for three metrics: total additions, total deletions, and files touched. This non-parametric test is appropriate given the skewed distributions observed in Figure 1.1. To complement this, I reported medians rather than means, as they better capture the typical scale of changes for skewed distributions. Together, the H-statistic and median values provide both evidence of significant differences and an interpretable summary of their magnitude.

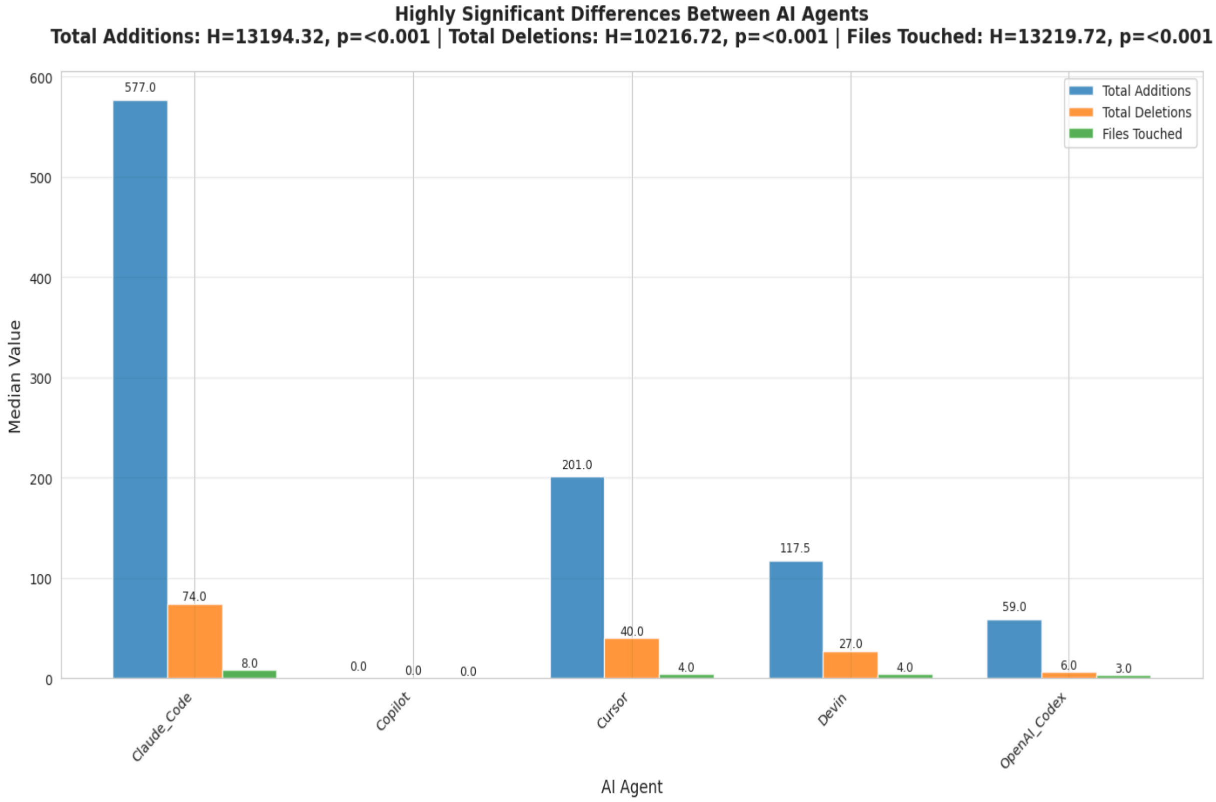


Figure: 1.2

The results (Figure 1.2) indicate highly significant differences between agents for all three metrics ($p < 0.001$ & high H value). Median comparisons reveal that Claude_Code shows the largest median additions and deletions, Copilot has extremely small median values, Cursor and Devin produce moderate to large changes, and Codex lies in between.

These findings confirm that agents do not behave uniformly. Copilot primarily produces small-scale edits, whereas Claude_Code and Devin frequently create larger and broader changes. This reinforces the need for agent-specific review practices: PRs from Copilot may require lighter review effort, while PRs from Claude_Code or Devin demand greater scrutiny.

To identify specific differences between AI agents' modification patterns, I conducted pairwise Mann-Whitney U tests with Cliff's Delta effect size measurements for the "files touched" metric.

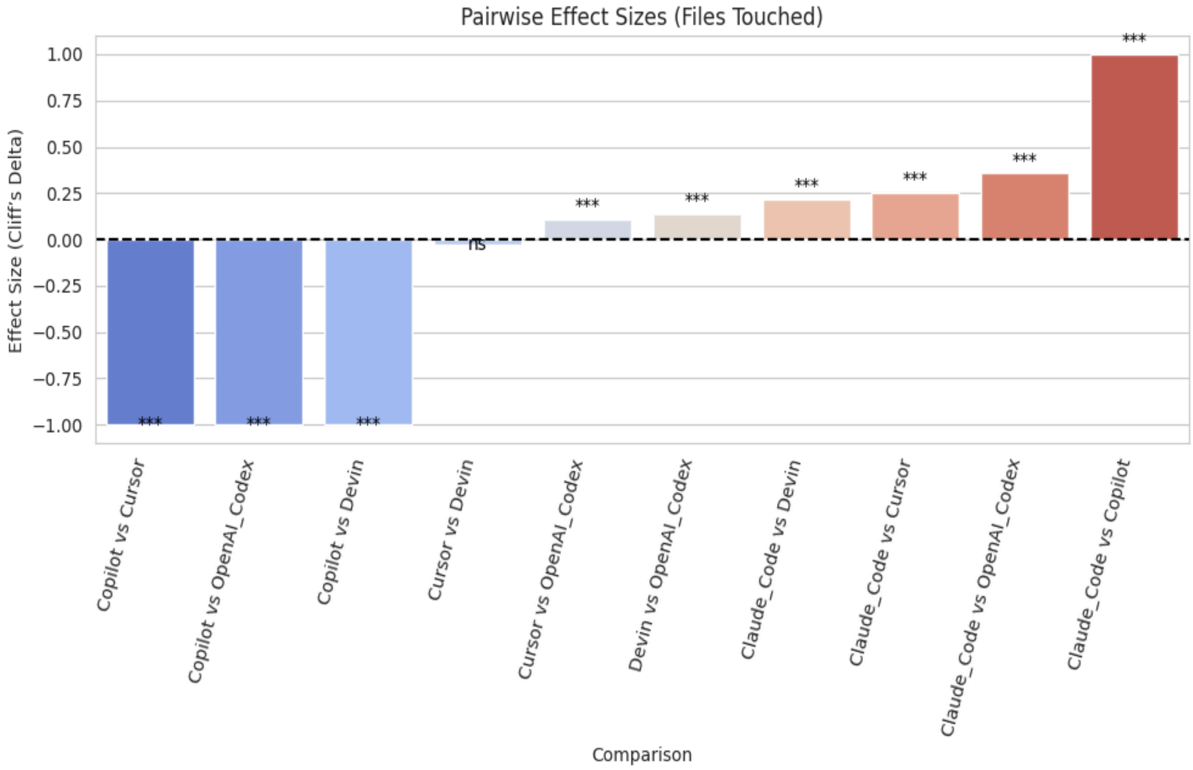


Figure: 1.3

Figure 1.3 visualizes these effect sizes in a bar plot, sorted by ascending delta for readability. The plot highlights the directional trends: negative bars (left side) show comparisons where the first agent touches fewer files, while positive bars (right side) indicate the opposite. Significance markers (***) are added above each bar, confirming that all differences are highly significant.

This pairwise analysis reinforces the overall Kruskal-Wallis findings, revealing a clear hierarchy in change scope: Claude_Code tends to change the most files per pull request, while GitHub Copilot is the most conservative, typically modifying the fewest files. The effect sizes between these two agents are substantial, confirming they have fundamentally different approaches. Devin and OpenAI Codex fall in the middle of this spectrum, with Cursor being closer to Copilot in its focused changes. This analysis will help developers to choose an AI agent based on whether they need broad, multi-file changes or focused, single-file fixes.

So the overall quantitative analysis reveals a clear spectrum of coding behaviors where GitHub Copilot is the most conservative (minimal, focused changes), followed by Cursor. OpenAI Codex and Devin show moderate scope, while Claude Code operates at the largest scale (extensive, multi-file modifications). This analytical hierarchy allows developers to select agents based on whether they need focused or targeted fixes or broader refactoring.

IV. Results [Qualitative Analysis]

To assess the qualitative alignment between PR descriptions and actual code changes, using the SentenceTransformers' all-MiniLM-L6-v2 model I computed cosine similarity scores between PR bodies (full_description) and code summaries (created code_summary).

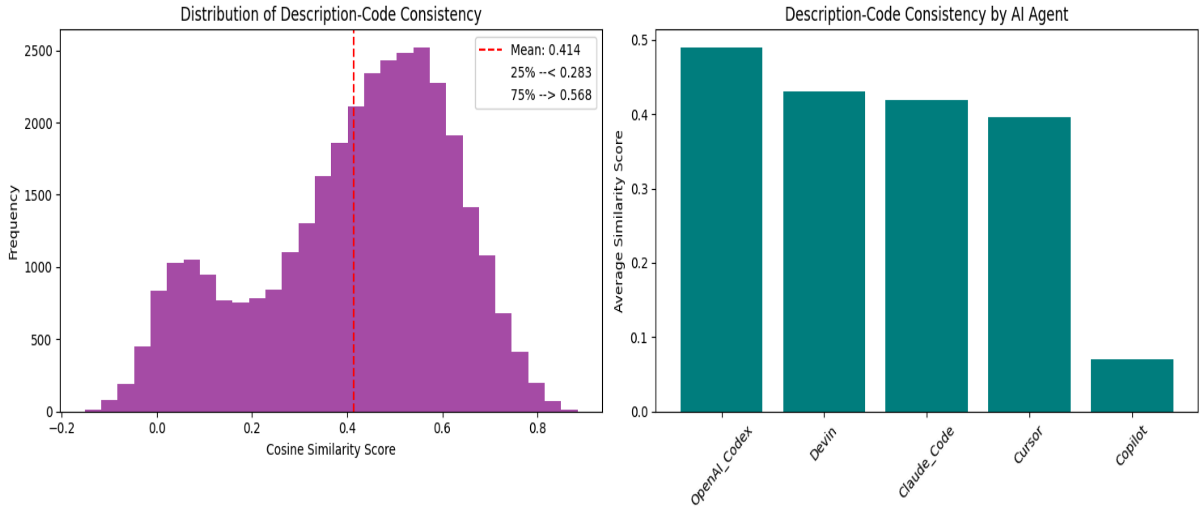


Figure: 2.1

The “Distribution of description-code consistency” (Figure 2.1, Left) shows a right-skewed pattern with a mean cosine similarity score of 0.414, indicating descriptions capture about 41.4% of the semantic content of the code changes. The peak frequency is around 0.4-0.6, with 25% of PRs having weak alignment (<0.284), while 75% show stronger consistency (>0.569), suggesting descriptions are often helpful but imprecise, with variability across agents. (Figure 2.1, Right) shows OpenAI Codex (about 0.5), Devin (slightly above 0.4), Claude_Code (0.4), Cursor (about 0.4), and Copilot (around 0.1), highlighting Codex’s strength and Copilot’s weakness, underscoring the need for improved agent performance.

Then to evaluate variations in description-code consistency across AI agents, I conducted an ANOVA on 33,580 Agentic-PRs from the AIDev dataset.

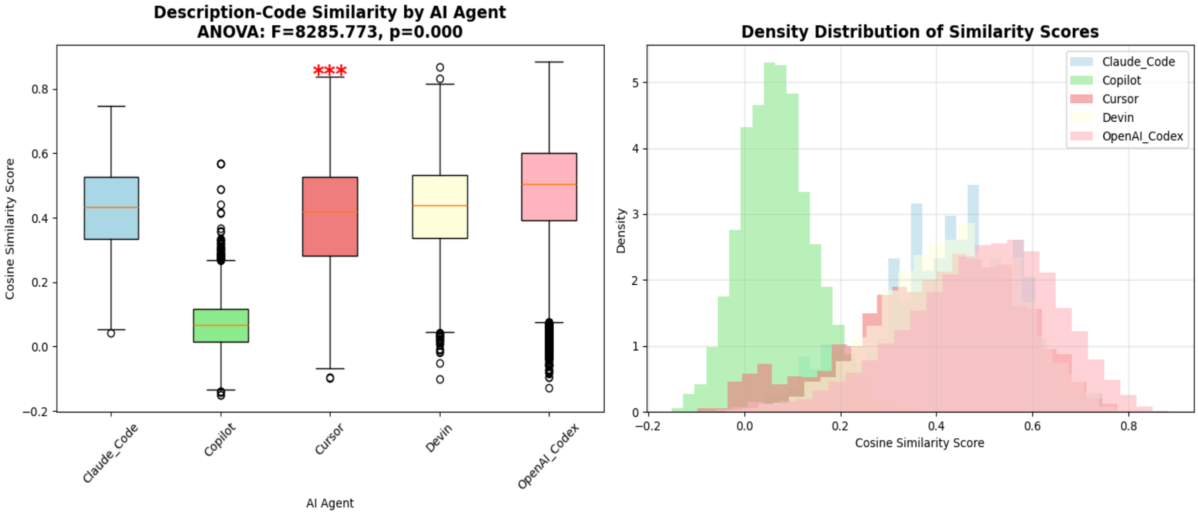
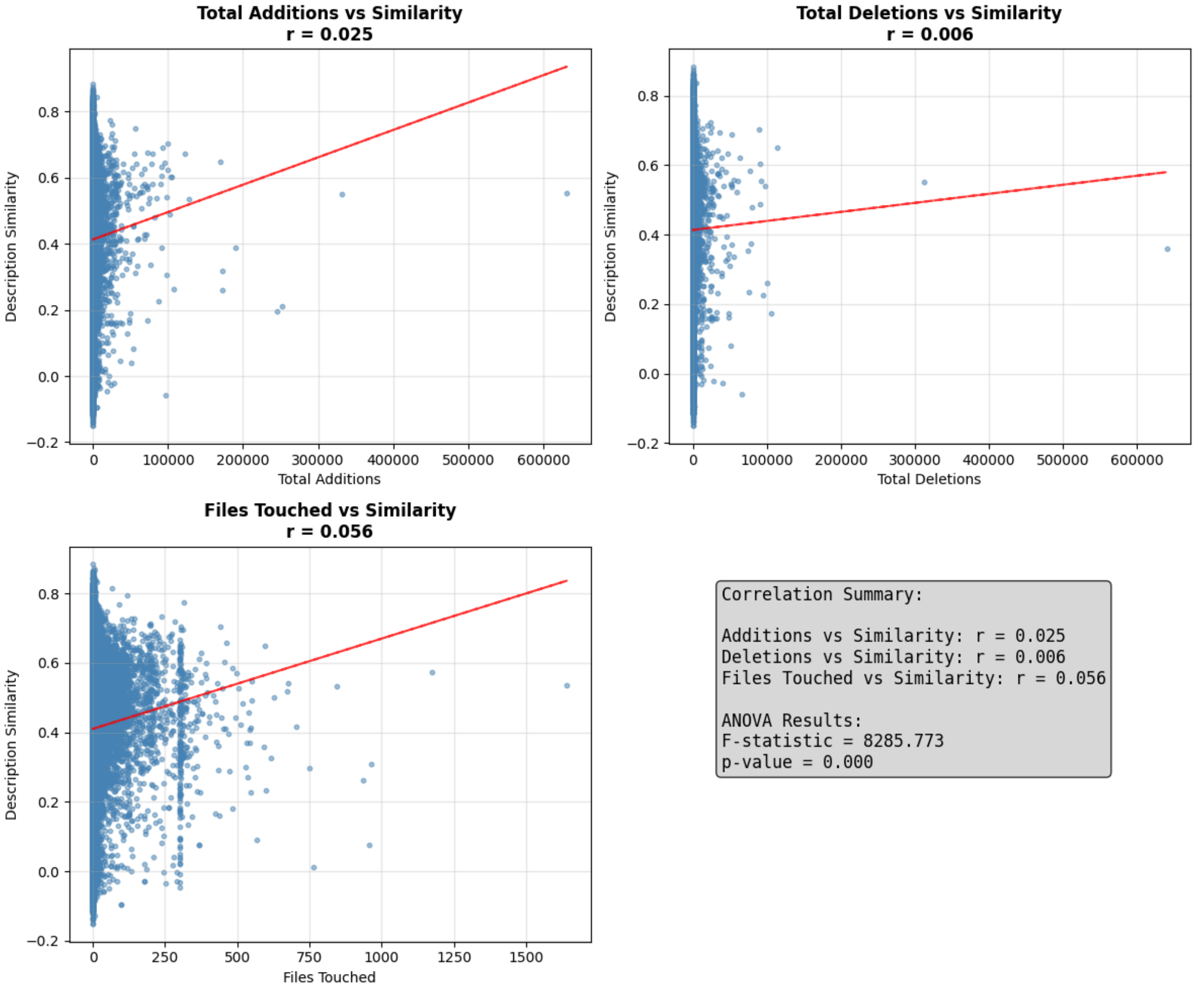


Figure: 2.2

Figure 2.2 (left) yields a significant F-statistic of 8285.773 (p=0.000), confirming distinct agent performance. Boxplots reveal OpenAI Codex with the highest similarity (near about 0.6), followed by Claude_Code, Devin, and Cursor (around 0.4-0.5), while Copilot lags (only around 0.15), supported by the density plot (Figure 2.2, right) showing overlapping distributions skewed toward moderate scores (0.2–0.6).

Finally, I explored the relationship between description-code similarity and code change metrics (total additions, deletions, files touched) across 33,580 Agentic-PRs from the AIDev dataset, using Pearson correlation and scatter plots.



```
Correlation Summary:

Additions vs Similarity: r = 0.025
Deletions vs Similarity: r = 0.006
Files Touched vs Similarity: r = 0.056

ANOVA Results:
F-statistic = 8285.773
p-value = 0.000
```

Figure 2.3

The analysis reveals weak correlations: total additions (r = 0.025), total deletions (r = 0.006), and files touched (r = 0.056), with a slight decline trend as change volume increases, indicating that larger PRs tend to have lower similarity. This suggests AI agents maintain almost similar consistency levels across small and large modifications, though a slight decline in similarity is observed for larger PRs.

So based on the qualitative analysis, the alignment between PR descriptions and code changes, measured via cosine similarity is moderate but varies significantly and AI agents maintain almost similar consistency levels across small and large modifications.

V. Discussion

The quantitative and qualitative analyses of 33,580 Agentic-PRs from the AIDev dataset provide a nuanced understanding of how AI coding agents shape software engineering workflows. My analysis reveals that AI coding agents exhibit distinct behavioral patterns in their pull request contributions.

Quantitatively, I observed a clear hierarchy in modification scope: GitHub Copilot produces the most conservative changes (minimal code modifications), while Claude Code operates at the largest scale with extensive, multi-file modifications followed by OpenAI Codex, Devin and Cursor. These differences are statistically significant ($p < 0.001$) and consistent across all code change metrics. The heavy-tailed distributions indicate that while most Agentic-PRs involve focused changes, AI agents are capable of substantial refactoring when required.

Qualitatively, the mean cosine similarity of 0.414 between PR descriptions and code summaries indicates moderate alignment, with 25% of PRs below 0.284 and 75% above 0.569, reflecting variability in agent accuracy. The ANOVA result ($F = 8285.773$, $p = 0.000$) confirms agent-specific differences, with OpenAI Codex outperforming (near about 0.5) and Copilot underperforming (around 0.1), while weak correlations suggest a slight decline in consistency with larger PRs. It indicates that description quality is an agent-specific characteristic rather than being influenced by change scope, pointing to opportunities for improved prompting or model training.

These findings enable strategic agent selection: Copilot for targeted fixes, Claude Code for broad refactoring, and intermediate agents for balanced tasks. The consistent description accuracy across change sizes provides confidence that AI agents maintain communication quality even for complex tasks, though very large modifications may require additional human verification.

Finally, it can be said that all these findings resonate with AIDev's emphasis on human-AI collaboration, but limitations exist. The embedding model (all-MiniLM-L6-v2) may miss domain-specific nuances, and the dataset's bias toward >100-star repositories could skew results. Future work could incorporate more capable embedding models to take domain-specific nuances into consideration along with exploring agent architectures or prompting strategies to enhance consistency, alignments.

VI. Conclusion

This study of 33,580 Agentic-PRs from the AIDev dataset demonstrates that AI coding agents transform software engineering with efficient, small-scale changes, though their impact varies by agent and task complexity. Three key findings– First, agents occupy distinct niches in the modification spectrum (Claude_Code for widespread changes, Copilot for targeted fixes) enabling developers to match tools to task needs. Second, description-code consistency differs significantly across agents, though a slight decline with larger PRs suggests minor instability. Third, the weak correlations with change size indicate that AI limitations primarily from semantic understanding, with scalability as a secondary concern.

These insights provide a foundation for strategically integrating AI teammates into workflows, as envisioned by AIDev. Future research should prioritize enhancing agent's impact for large-scale changes and developing task-based selection frameworks. As AI agents evolve, understanding their unique strengths will be vital for optimizing human-AI collaboration in SE 3.0.