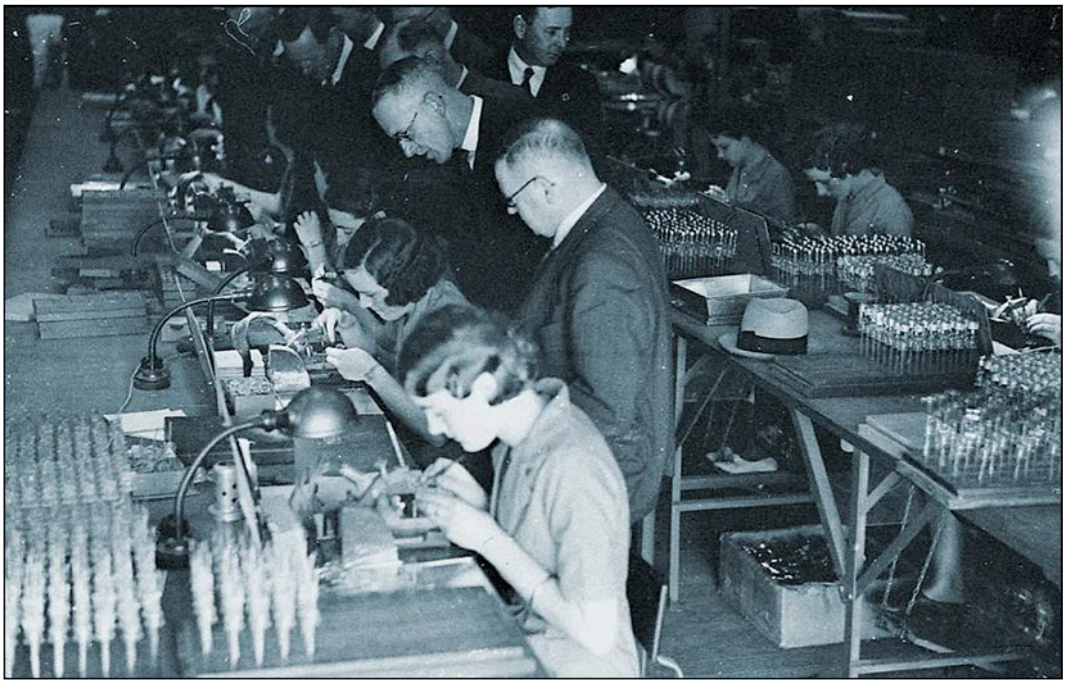
Downgrading From Claude Fable Barely Affected Dev Productivity During Ban

Originally published Weds 18 Nov 2026 at https://www.unite.ai/downgrading-from-claude-fable-barely-affected-dev-productivity-during-ban/



It is very difficult to accurately assess whether new AI models have improved their performance over their predecessors or not – or even to define what a productivity improvement necessarily constitutes, since new releases may demonstrate improvements in one area, but degraded performance in another, in much the same way that conventional software and frameworks can. Sometimes these are voluntary, human-chosen tradeoffs, sometimes 'collateral damage'.

That's assuming 'fair play'. There are plenty of cases where developers have rigged software to perform better in benchmarks, sometimes making it recognize when it's being tested and temporarily change how it operates to get a better score. AI models too can recognize when they're being evaluated, and behave differently, even without being deliberately designed to do so.

In search of more meaningful and honest metrics, developers often establish their own casual yardsticks, such as Simon Wilson's qualitative Pelican riding a bicycle challenge, wherein Vision Language Models (VLMs) are tasked with creating something that is absent from the training data*, and often perform poorly – though things have improved somewhat throughout 2026:



Simon Wilson's metric asks a VLM to picture a Pelican riding a bicycle. Lately, AI has gotten much better at this, compared to the hilarious efforts of around 18 months ago. But some suggest that this may be because the challenge is now enshrined in the training data. Source

However, a benchmark can only measure the particular skills it was designed to test, and even when the results are reliable, it remains unclear how much those skills *matter* in everyday use, or whether a higher score will translate into more work being completed.

In the case of coding models, that can be investigated by looking at what happens to developers' output when they switch from one model to another.

© Martin Anderson and/or commissioning entities

Doing [the Same] with Less

In June this year, the 19-day White House ban on the use of Claude Fable provided researchers with an unusually clear testing environment, as developers submitting pull requests (PRs) to GitHub downgraded to less powerful versions of Claude. Since all versions of Claude co-sign PRs, and since that information is searchable via the GitHub API, the data is freely available.

This opportunity was taken up in a new study from Harvard University, in which the work of 297 Fable 5 users was tracked before, during and after the ban, and compared with that of 8,908 developers who continued using the older Opus 4.8 model.

Fable 5 had previously been awarded a score nearly 11 points higher than Opus 4.8 on the SWE-Bench Pro coding benchmark. In the Harvard study, however, access to Fable 5 was associated with an increase of *just 1.4 percentage points* in the proportion of submitted code changes accepted into GitHub projects within seven days, a difference the paper claims to be practically a rounding error; and a wider analysis also found no significant productivity gains from upgrading to other newer AI models:

> *'Results show that having access to Fable 5 changed the merge rate by only 1.4 percentage points, and the change was not statistically [significant]. We also analyzed five of the eight model upgrades in 2026 and found that the merge rate did not rise after these upgrades [either].*
>
> *'Similarly, developers did not get more pull requests merged per week. These results show that gains on current coding benchmark did not really translate into more finished work.'*

Across five upgrades in 2026, involving 1,208 developers and more than 15,000 proposed code changes, no statistically significant improvement was found in the rate at which submitted code was merged, or in the number of changes successfully completed each week.

Though the paper acknowledges that GitHub merges are an imperfect measure of productivity, the ban offered a very rare opportunity to assess the effects of switching models under real working conditions, in the context of a sector that attracts huge global interest, but which operates under the maximum possible corporate opacity.

Acknowledging the difficulty of establishing empirical criteria in such cases, the new work[†] observes:

> *'People who use new models first may be more experienced, so when we compare productivity across developers, we cannot tell if the difference comes from the model or from the people using it. Even if we observe the same people, we do not know if they give the same tasks to different models.*
>
> *'For example, some people might use Fable 5 for the hardest tasks and switch to lower-tier models for easier tasks. If they give the harder tasks to the strongest model, the strongest model may look less productive on almost any measure we choose, since harder tasks take longer and are less likely to be merged.'*

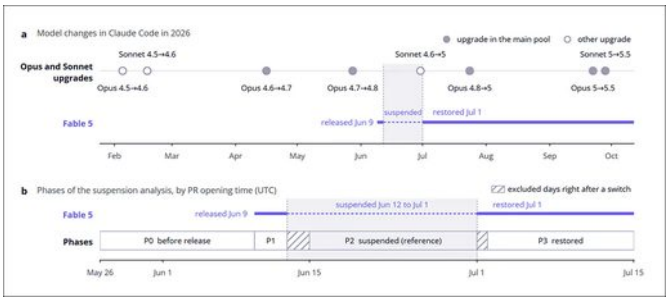
The paper also contends that previous studies have been skewed by the fact that most code changes came from a small number of projects, each with its own rules for accepting contributions; and that these differences have made it difficult to establish whether productivity gains were actually caused by upgrading to newer models.

Method

For the main experiment[††], the GitHub pull requests of 297 developers identified in the study as Fable 5 users were divided into four periods: before the model's June 9 release; during its three days of availability; throughout the 19-day suspension; and after access was restored on July 1.

Their merge rates were compared with those of 8,908 developers identified as using the older Opus 4.8 model, allowing changes associated with the suspension to be distinguished from wider fluctuations in GitHub activity.

The comparison assumed that, without the ban, both groups would have experienced similar changes in their merge rates. Changes in merge rates were compared between the two groups, with differences between individual developers, and day-to-day fluctuations, accounted for in the statistical analysis:



Timeline of Claude model upgrades and the Fable 5 suspension. The upper panel shows eight Opus and Sonnet upgrades, with filled circles marking the five examined in the wider study. The lower panel shows the four periods used to assess how the suspension period affected PR merge rates, with the ban serving as the reference period. Purple lines show when Fable 5 was available or suspended; gray shading marks the ban; and hatched sections indicate days excluded immediately after access was suspended or restored. Source

© Martin Anderson and/or commissioning entities

In regard to data, public GitHub PRs created with Claude Code were identified through the GitHub Search API, using the 'Generated with Claude Code' notice automatically included in their descriptions. Where the specific model name had also been copied from the code's commit history into the PR description, it was possible to establish the version of Claude used.

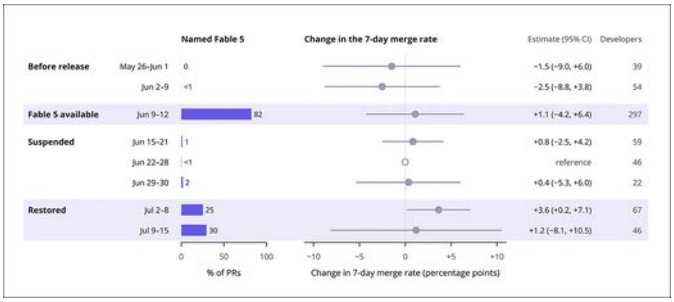
PRs submitted around the Fable 5 suspension and subsequent Claude upgrades were collected between November 2025 and October 2026, with at least four weeks allowed for the latest submissions to be merged.

The 297 developers who used Fable 5 during its brief availability were selected for the main study group, while 8,908 developers who used Opus 4.8, with no recorded use of Fable 5 served as the control group.

The main metric of productivity was the *proportion of submitted code changes accepted into GitHub projects within seven days*, allowing the rate of successful submissions to be compared across the periods before, during and after the suspension. Additional checks covered PRs merged within an hour; PRs closed without being merged; the number of comments received; changes subsequently reversed; and failures in GitHub's automated checks.

Tests

The loss and subsequent restoration of Fable 5 access had little measurable effect on developer productivity, with access to the newer model associated with an increase of just 1.4 percentage points in the proportion of PRs merged within seven days; a difference deemed 'not statistically significant':



Comparison of Fable 5 usage and PR merge rates before, during and after the ban. Left: the percentage of PRs produced using Fable 5. Right: changes in the seven-day merge rate relative to June 22–28, with dots showing estimates and horizontal lines indicating 95% confidence intervals. Shaded rows mark periods when Fable 5 was available. Despite large changes in model usage, little change in merge rates was observed.

Additionally, across five Claude model upgrades, involving 1,208 developers and more than 15,000 code submissions, no improvement in the proportion of code accepted into GitHub projects was found. The rate actually fell slightly, by an estimated 0.9 percentage points.

Developers using newer models were also found to complete just 2.4% more accepted submissions per week – a difference too small to establish a reliable improvement. Even where benchmark scores had risen by between 5 and 18 points, no significant productivity gains were observed.

The newer models were also found to perform no better in GitHub's automated code checks, with failure rates remaining essentially unchanged.

The paper concludes:

> *'We hypothesize that the main reason Fable 5's gains on benchmarks did not translate to real-work productivity is that the tasks developers give agents are already within what the older model can do.'*

And continues:

> *'Our findings add evidence from real work to what prior work has warned about: a benchmark score means little to people choosing a model without evidence that it measures the work it is used to [describe].*
>
> *'If regulators and buyers judge how much stronger a model is by its benchmark [score], the difference they rely on will likely not show up in user productivity.'*

Conclusion

Consistent, year-on-year metrics have long been a pre-requisite for the business aspect of industrial systems; but they are particularly hard to replicate in real and accurate terms, in the field of AI. Part of the reason for this is the speed with which the 'test data' that defines a benchmark *enters the training data of the next model* and becomes useless for testing purposes. This alone pretty much breaks the template for corporate benchmarking, and for consistent scientific testing, where researchers depend on breaking formerly-established metrics that may now be meaningless.

Indeed, it has been amazing over the last ten years to watch the number of Computer Vision research papers that still adhere to long-outmoded standards such as ImageNet, simply because their provenance goes back so far.

© Martin Anderson and/or commissioning entities

The results of this paper present *developers* in a rather good light, as adaptable and flexible proponents of whatever system is available and running; the fact that they perform at much the same speed with either the latest or the last AI indicates that they are optimizing to durations and deadlines, regardless of the tools to hand.

* *As one recent Hacker News commenter pointed out, this is probably no longer true, since the 'Pelican on Bicycle' data is now in training data, and the test is therefore compromised.*

† *Titled 'Higher-Scoring Coding Models Did Not Get More Work Done'.*

†† *Two other questions are posed in the study, but I do not feel they are particularly germane to the topic, or comparable in significance to the primary question, and have not reported on them here. Please refer to the source PDF for further details.*

First published Wednesday, November 18, 2026

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Website: martinanderson.ai
Contact: martin@martinanderson.ai