

Supervised Practice Time and the AI-Code Workflow: Evidence from Finance Undergraduates at a Regional Chinese University

Yichao Mo

Yc.M@tlu.edu.cn

Tongling University

Kexin Ouyang

Southwestern University of Finance and Economics

Article

Keywords: Supervised practice time, AI-code workflow, Generative AI in education, Finance education

Posted Date: October 12th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10999692/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Supervised Practice Time and the AI-Code Workflow: Evidence from Finance Undergraduates at a Regional Chinese University

Abstract: Generative AI now produces runnable code from plain-language prompts, moving the question in quantitative finance courses from whether students can write code to whether they can command the AI-code workflow of prompting, running, and revising. We study how supervised practice relates to that command in four intact classes of 138 finance undergraduates at a regional Chinese university, randomized at the class level to a bring-your-own-device arrangement or a standard schedule, with the instructor, syllabus, and AI assistant held constant. More supervised practice went with stronger command of the workflow. Over four MATLAB assessments the classes with more supervised practice opened an advantage of about eleven points from a level start, and a second-semester Stata assessment pointed the same way. Two questionnaires locate the reason in the students themselves, who name weak self-initiative as their main barrier, so those class hours carry practice that would otherwise not happen. Once producing runnable code no longer binds, judging whether the answer is correct becomes the difficulty they report most.

Keywords: Supervised practice time; AI-code workflow; Generative AI in education; Finance education

1. Introduction

Generative AI now turns plain-language prompts into runnable code, and writing syntax is no longer the barrier to entry in quantitative finance courses. What binds instead is the AI-code workflow of prompting, running, and revising, so the demand moves from producing code to directing and checking a system that produces it. This raises a question for teaching: how do finance students come to command the AI-code workflow?

The AI-code workflow is a skill learned by doing. Students have to prompt, run the returned script, inspect the output, and revise the prompt until the result holds up. Yet many finance students do not rehearse that sequence on their own after class. We use supervised practice time (SPT) to mean instructor-present class time in which a student has a computer available to run that sequence. The question is whether more SPT is associated with stronger command of the AI-code workflow.

We take this question to a two-semester cluster-randomized study in four intact finance classes. The design holds the instructor, syllabus, materials, and AI assistant fixed while varying, at the class level, whether students had a machine in front of them during regular sessions. In the first semester two classes worked on their own laptops throughout and two did not, and every class sat four hands-on MATLAB assessments on financial-market problems. In the second semester we followed the two lower-SPT classes further, one drawing a further round of the arrangement and the other none, and both took a Stata assessment on a corporate-finance problem as an exploratory check on whether the gain travels to new software. Two post-course questionnaires closed the study, one for each pair of classes, and 127 of 138 responses were analyzable.

We find that more supervised practice goes with a widening hands-on advantage, and a second-semester assessment in new software points the same way. In learning the AI-code workflow, most students name weak self-initiative as their greatest obstacle, which fits a reading in which those who will not practice alone gain most from supervised hours. In using the workflow, once producing runnable code no longer binds, judging whether the AI's output is correct is the step at which they most often get stuck.

This study contributes to two strands of literature. The first asks whether access to computing raises attainment, and its estimates do not converge. Angrist and Lavy (2002) find no detectable gain from classroom computers and a negative effect on fourth-grade mathematics, Cheung and Slavin (2013) report a mean effect near 0.15 standard deviations for educational technology in K-12 mathematics, Zheng et al. (2016) find positive but small effects across one-to-one laptop programs, and Sung et al. (2016) report a moderate mean effect across studies of mobile devices, with the spread traced to how devices were used rather than to whether they were supplied. Provision is the treatment throughout, and the dispersion is itself the opening. What stays unsettled is how device use should be organized once provision is no longer the constraint. We hold access constant and vary only the class hours in which students worked at a machine with the instructor present, and we find that this placement of machine time tracks command of the workflow.

The second strand establishes that what is done with class time matters. Time spent working outperforms time spent listening (Freeman et al. 2014), active learning narrows attainment gaps across student groups (Theobald et al. 2020), extending instruction time raises achievement in a cluster-randomized setting (Andersen et al. 2016), and a fixed amount of practice yields more distributed than massed, in general learning and introductory programming alike (Cepeda et al. 2006; Li et al. 2021).

This work was carried out where students produced the solution themselves, and it does not tell us what counts as practice once a language model supplies code that runs. The generative-AI literature has not settled the question either. Kasneci et al. (2023) is a commentary on opportunities and risks, Kazemitabaar et al. (2023) run a controlled experiment with novices aged ten to seventeen over a short exposure, and Prather et al. (2024) observe novice programmers in the laboratory. We treat the unit of practice as the full prompt-run-inspect-revise cycle, and supervised hours track performance on tasks scored at the end of that cycle. What students report as the standing difficulty is not producing a script but judging whether its output is right.

The rest of the paper proceeds as follows. Section 2 describes the institutional setting and design; Section 3 lays out the empirical strategy and reports the main and exploratory evidence; Section 4 discusses mechanism, cross-software transfer, and the remaining bottleneck; and Section 5 concludes.

2. Setting, Design, and Measurement

2.1 Institutional context and class-level randomization

The study was conducted in the undergraduate finance program of a regional Chinese university with limited computing provision. The program has no dedicated finance laboratory, and students reach institutional machines only in scheduled sessions in shared computer rooms. Software-based teaching therefore relies either on those scheduled sessions or on students' own devices.

We ran assignment at the class level before teaching began. A single cohort formed four administrative classes of 35, 30, 32, and 41 students. In the first-semester, ahead of the first course meeting, we seeded the random-number generator with the first teaching day's date and drew two of the four classes into the bring-your-own-device (BYOD) condition, leaving the other two as the control condition; the two drawn are reported here as classes 1 and 2 and two others as classes 3 and 4. For the second-semester follow-up we repeated the procedure with a seed set to the first teaching day of the Corporate Finance course, drawing one of classes 3 and 4 into a BYOD condition, reported here as class 4. The study was not preregistered, and the assignment script, seeds, and log were fixed before any teaching or measurement. This design is a cluster-randomized study with four clusters.

Class 4 entered through a separate vocational channel rather than the general college entrance examination. Its students were ranked on a cultural-literacy examination scored out of 300 combined with a vocational-skills examination scored out of 450. This composite sits on a different scale from the general admission score, so class 4 cannot be placed on the same attainment metric as classes 1 to 3,

and the four clusters are not exchangeable on admission channel. We show in the robustness analysis that dropping class 4 leaves the main hands-on estimate essentially unchanged.

All students used DeepSeek as the AI assistant. Students in mainland China cannot directly reach ChatGPT, Claude, or Gemini, so DeepSeek was the workable choice. All classes used the free web interface to DeepSeek in its default configuration, in the autumn semester of 2025 and the spring semester of 2026. DeepSeek accepts Chinese-language prompts, runs on domestic infrastructure, and produced the MATLAB and Stata code the courses required. The provider updated the underlying model during the study without notice, and any such change applied to all classes in the same weeks. Holding the AI assistant constant across all four classes leaves the BYOD arrangement as the contrast between conditions.

Throughout the paper, the two first-semester conditions are called the treatment and control groups. The second-semester follow-up is reported by class label, since each condition there contains a single class, and all questionnaire results are reported by class label.

2.2 The intervention and supervised practice time

The first semester delivered the intervention through MATLAB and Financial Model Analysis. The course ran for 48 class periods of 45 minutes over 16 teaching weeks, in sessions of two consecutive periods. Two-thirds of the periods were held in a general teaching room and one-third in a shared computer laboratory. Each class met weekly in the teaching room and entered the laboratory every second week, giving every class the same laboratory allocation, with one treatment class and one control class in each weekly laboratory slot.

The two conditions differed in supervised practice time, which is the instructor-present class time in which a student had a computer available for practice. Treatment students brought their own laptops from the second teaching-room session onward and worked along with the instructor, while control students had a machine only during laboratory sessions and otherwise watched the same instructor-led demonstration. Both conditions shared the same laboratory hours, so the difference came entirely from the teaching-room sessions, where treatment students had a machine to practice and control students did not. This difference in SPT is one of scheduled machine time, since we recorded attendance by session but not how each session split between demonstration and student practice.

Both conditions received the same teaching resources. Before each new unit, the worked code, the slides, and a recorded walkthrough were posted to each class's course group on WeChat. In class the instructor ran the same slow, step-by-step demonstration with concurrent explanation for every class;

treatment students reproduced it on their own machines, while control students watched. No class received extra guidance on AI use or MATLAB operations, and control students were free to bring a laptop but generally did not.

The second semester followed the two lower-SPT classes, classes 3 and 4, into Corporate Finance. Most sessions were held in a classroom, and a short block of laboratory sessions near the end of the term taught and practiced basic Stata operations. Both classes shared that block, and the single Stata assessment fell within it. Class 4 also worked on its own devices during regular fortnightly sessions under a second BYOD requirement, which added further SPT, while class 3 kept only the shared laboratory hours.

2.3 Outcome Measures

Learning was tracked with hands-on tests in both courses. In MATLAB and Financial Model Analysis, four tests ran across the term, each asking students to complete four short MATLAB tasks scored in 25-point increments to a 100-point total. In Corporate Finance, a single Stata test with four tasks closed the follow-up, scored on the same 25-point rule.

A fixed grading rubric was shared across all four classes. Each of the four tasks received either 25 or 0 points under a single tolerance rule. Full marks required that the AI-produced script ran without errors and returned a numerical result within 0.01 of the reference value. Every task was set to resolve to such a result, and no task was graded on the appearance of a table or a figure. Any other outcome scored zero. Students submitted a short report containing the final prompt, the AI-generated script, and the console output it produced. The course instructor graded every script against a prepared answer key. The rule is binary and mechanical, which limits rater judgment. The instructor graded unblinded to condition, with no second rater and no automated scoring.

Students were told before each assessment that any genuine attempt would receive full credit for the laboratory component of the course, regardless of whether the answer was correct. The research score was recorded separately from the course grade, which reduced the incentive to copy or to obtain tasks in advance. Students who could not attend on time completed the assessment remotely and submitted their work within one week. In semester 1 this covered one student in class 3 for T2 and three students for T3, two in class 2 and one in class 4; in semester 2 it covered one student in class 3. Every student thus has a score at every wave.

The four tasks in each test ran from routine description to multi-step verification. The same difficulty pattern repeated across T1 to T4 with course-appropriate content. A sample T4 set illustrates the levels students faced by the end of the term.

Level 1. *A 3-year coupon-bearing Treasury bond has a face value of 100 yuan and a 3% annual coupon paid once a year. Given a current yield to maturity of 2.5%, compute the theoretical price of the bond and judge whether it trades at a discount, at par, or at a premium.*

Level 2. *Use closing prices from 000852.xlsx, compute the daily log returns of the CSI 1000 Index, report their mean and standard deviation, and from these derive the annualized return and annualized volatility.*

Level 3. *Use closing prices from 000001.xlsx and 000300.xlsx, compute the daily returns of the Shanghai Composite Index and the CSI 300 Index respectively, and compute the correlation coefficient between the two return series on their common trading dates.*

Level 4. *Use 000852.xlsx and USDCNY.xlsx, analyze the 2015 correlation between returns on the CSI 1000 Index and the USD-CNY exchange rate, where date formats differ across the two files and occasional missing values must be diagnosed and resolved before computation.*

The Stata follow-up test mirrored this four-level design in a corporate-finance setting. Both former control classes sat a common software orientation and then took this same graded test. Its four tasks are listed below.

Level 1. *A 3-year corporate bond has a face value of 100 yuan and a 4% annual coupon paid once a year, with principal repaid at maturity. Given a current yield to maturity of 3.5%, compute the theoretical price in Stata and judge whether it trades at a discount, at par, or at a premium.*

Level 2. *Import a corporate-bond dataset from Excel into Stata, and for the full sample report the mean, standard deviation, minimum, and maximum of the coupon rate.*

Level 3. *Merge two firm-level files by security code, keep records dated 2021-12-31, construct firm age as 2021 minus the founding year, and report the number of matched firms and their mean age.*

Level 4. *Convert string start dates to quarterly variables in two files, merge the bond micro-data with macro economic variables such as year-on-year growth on the year-quarter key, and save one analysis-ready dta file.*

All four assessments were taken in the shared computer laboratory on institutional machines, one assessment per laboratory session, with the full two periods available. Students had internet access and used DeepSeek during the assessment, which is why the submitted report includes the prompt. Sessions

were not invigilated and students worked individually. The laboratory slots alternated by week, so the four classes sat each assessment in different weeks on the same task set.

2.4 Questionnaires

Two questionnaires closed the study, one for each pair of classes. The first went to classes 1 and 2 and carried 13 single-choice items, while the second went to classes 3 and 4 and carried 15 single-choice items. Item order and option order were fixed. Both instruments opened at the end of the second semester, about six months after the MATLAB course had ended, within a few days of each other. Responses were collected anonymously.

Records were screened for straight-lining before analysis. We removed four responses that selected the first listed option on every item after the class-identifier question, one from the first instrument and three from the second, and all three of the latter came from class 3. All four were completed in well under a minute. This leaves 58 analyzable responses from the 65 students in classes 1 and 2 and 69 from the 73 students in classes 3 and 4, or 127 of 138 students overall. All questionnaire percentages below use these post-screening denominators.

The two questionnaire groups had different course histories when they answered. Classes 1 and 2 had taken the MATLAB practical course with the study instructor; their corporate finance course was taught by an outside instructor and included no AI practice. Classes 3 and 4 had taken both courses with the study instructor and followed the same demonstrations, and class 4 additionally worked on its own machines under the semester 2 arrangement. The questionnaires are used descriptively and were answered after the performance outcomes were measured. The main text reports the items that speak to the questions asked here. The full instruments and item-level frequencies for every item appear in Appendix Tables A2 and A3.

3. Evidence on Supervised Practice Time

This section draws on two semesters of hands-on assessments. Section 3.1 reports the first-semester comparison across four MATLAB assessments as the main analysis. Section 3.2 reports the second-semester Stata follow-up as exploratory, and Section 3.3 checks how far the main estimate depends on any single class. The design randomizes only four classes, too few for conventional significance tests, so we read the evidence through effect sizes and distributions.

3.1 Hands-On Evidence from Semester 1

We estimate the group difference with an analysis of covariance, given in specification (1), and read β as the quantity of interest.

$$Y_i = \alpha + \beta \cdot Group_c + \gamma \cdot T1_i + \delta \cdot Female_i + \varepsilon_i \quad (1)$$

Here Y_i is score of student i on a given hands-on assessment, $Group_c$ is the treatment indicator for the class c to which student i belongs, $T1_i$ is the baseline hands-on score, $Female_i$ is a sex indicator, and β is the group coefficient of interest. Standard errors are clustered at the class level. We do not enter class indicators alongside $Group$, since the four classes are perfectly nested within the two conditions and a matrix holding both is singular.

On T1 the treatment group averaged 8.46 points and the control group 7.53 points, a gap of under one point on a 100-point scale, and Table A1 shows the two groups were also close on sex composition. Every T1 score was either 0 or 25, since 22 of 65 treatment students and 22 of 73 control students completed exactly one task and the rest none, so the measure separates students only coarsely. The comparison rests on the class-level randomization described earlier, and unobserved differences in motivation, admission channel, or class climate remain open. The gap opens at the second assessment and holds through the fourth. The unadjusted mean difference runs 9.37 points at T2, 12.29 at T3, and 11.17 at T4, and conditioning on the T1 score and sex leaves them at 9.04, 11.86, and 10.88. Cohen's d tracks the same widening, from near zero at T1 to 0.61 at T2, 0.72 at T3, and 0.64 at T4.

Table 1. Hands-on assessment scores by group.

Outcome	Treatment	Control	Mean Diff	Adjusted Diff	Cohen's d
T1, Weeks 1-2, baseline	8.46 (11.92)	7.53 (11.55)	0.93	—	0.08
T2, Weeks 5-6	29.23 (15.03)	19.86 (15.57)	9.37	9.04	0.61
T3, Weeks 9-10	46.54 (17.61)	34.25 (16.41)	12.29	11.86	0.72
T4, Weeks 15-16	51.92 (16.12)	40.75 (18.40)	11.17	10.88	0.64

Note: The table reports group means and standard deviations for each hands-on test. The four tests T1 to T4 are each scored 0 to 100. The treatment group contained 65 students and the control group 73. Mean Diff is treatment minus control. Adjusted diff is the group coefficient from specification (1), conditioning on the T1 score and sex. Cohen's d is the unadjusted between-group standardized mean difference computed with a pooled standard deviation.

Figure 1 shows the gap forming at the bottom of the scale in T4. The control group leaves 31 of its 73 students at 25 or below, against 9 of 65 in the treatment group. Both groups peak at 50. The shift is

largest at the bottom of the scale, where the control group leaves far more students at one task or none, though the top also gains, with 12 of 65 treatment students at 75 or 100 against 6 of 73 in the control group.

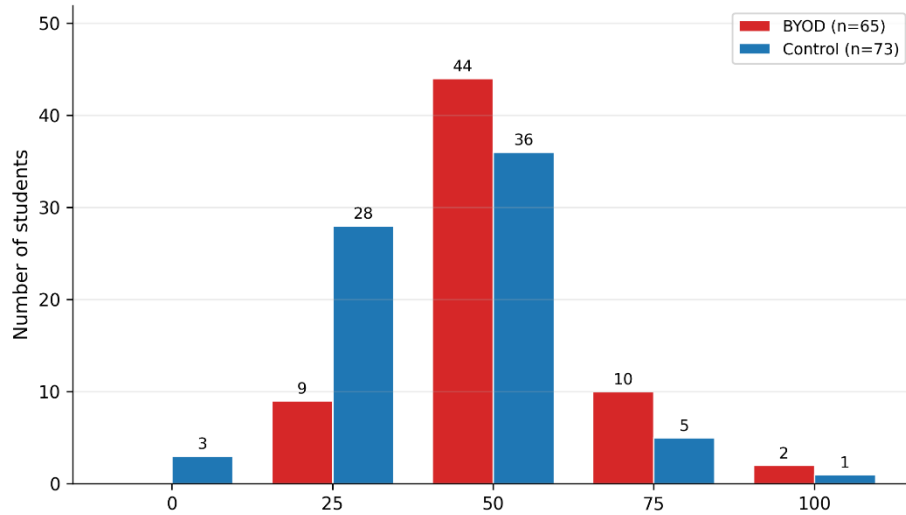


Figure 1. Distribution of T4 hands-on scores by group.

Note: Counts of students at each of the five possible totals, 0, 25, 50, 75, and 100, each corresponding to the number of the four tasks completed. Treatment 65 students, control 73 students.

We further estimate a linear mixed-effects growth model, given in specification (2), which tracks how the hands-on gap widens across the four tests. It pools all four scores per student and lets the treatment effect grow with each test.

$$Y_{it} = \alpha + \pi \cdot Wave_t + \lambda \cdot Group_c + \delta \cdot Wave_t \times Group_c + \varphi \cdot Female_i + u_i + \varepsilon_{it} \quad (2)$$

Here Y_{it} is student i 's score on test t , and $Wave_t$ runs from 1 to 4 across the four tests. $Group_c$ and $Female_i$ are indicators, u_i is a student random intercept, and ε_{it} is the residual. The coefficient δ is the extra growth per test in the treatment group. We fit the model by restricted maximum likelihood and cluster standard errors at the class level.

Table 2 reports the growth model in full. The long-format regression pools 552 student-test observations and clusters standard errors at the class level. The Wave term shows both groups improving by about eleven points per test, a common time trend across conditions. The Group coefficient of 0.10 comes from fitting the divergence as a straight line. Extrapolated to T1, that line implies a group difference of 3.47 points, above the observed 0.93, which is why the Group term is not the baseline difference. The Wave-by-group interaction of 3.37 points is the quantity of interest, the extra gain the treatment group makes per test, and it matches the widening gap in the raw means. The observed gap widens fastest from T1 to T3 and then levels off.

Table 2. Wave-by-group growth model on the four hands-on tests

Term	Coef
Wave	11.40
Group	0.10
Wave $\times$ group	3.37
Female	-1.57
Intercept	-2.12

Note: The outcome is the hands-on test score on the 0 to 100 scale, pooled over T1 to T4. The unit of observation is the student-wave pair, with 552 observations from 138 students. Wave indexes the four hands-on tests T1 to T4, coded 1 to 4. Group is the treatment indicator. Wave enters linearly, so Wave $\times$ group is an average gain per assessment, not a description of the shape of the divergence.

Table 3 summarizes the distribution of individual learning slopes behind that interaction. Each student's slope is the within-student regression of the four hands-on scores on the wave index. The growth advantage spreads across students and is not carried by a fast-learning minority. Every treatment-group student has a positive slope, against 91.8 percent of control students, and 87.7 percent of treatment students gain at least ten points per assessment against 63.0 percent of control students. The median treatment student gains fifteen points per wave, five points above the median control student.

Table 3. Distribution of individual learning slopes across the four hands-on tests.

Group	N	Mean (SD)	Median	Min	Max	Share positive, %	Share at least 10, %
Treatment	65	14.77 (5.71)	15.00	2.50	32.50	100.0	87.7
Control	73	11.40 (6.36)	10.00	0.00	30.00	91.8	63.0

Note: The individual slope is the within-student ordinary-least-squares slope of the four hands-on test scores on the wave index from 1 to 4, in points per wave. Share positive is the percentage of students whose slope exceeds zero, and Share at least 10 is the percentage whose slope reaches ten points per wave.

3.2 Cross-Software Evidence from Semester 2

In the semester after MATLAB and Financial Model Analysis, we asked whether added SPT separates the two former control classes. A second class-level draw sent class 4 to work on its own devices during regular sessions, while both classes shared the computer-room time for Stata instruction near the end of the term. The end-of-course questionnaire asked whether students had used Stata before the Corporate Finance course began, and class 4 reported the lower share, 12.5 percent against 20.7 percent in class 3. Before screening the shares were 12.5 and 28.1 percent; all three removed responses

came from class 3 and all had reported prior use, so the screening reduced this baseline difference. The two classes had tracked each other closely through the previous semester, with near-identical average growth across the four MATLAB assessments, 11.71 and 11.02 points per assessment, and by T4 class 4 stood 4.4 points ahead.

Figure 2 shows class 4 outperforming class 3 across the assessment. Panel A shows its distribution shifted right, with 48.8 percent of class 4 scoring 75 or 100 against 15.6 percent of class 3, while class 3 piles up at 25. Panel B traces the four tasks. The two classes nearly tie on the first at 95.1 against 90.6 percent, then separate on the remaining three, 82.9 against 43.8 percent on the second, 31.7 against 18.8 percent on the third, and 36.6 against 9.4 percent on the fourth. The gap opens as soon as a task requires several chained operations. Class 4 averaged 61.59 points against 40.62 in class 3, a difference of 20.97 points and a Cohen's d of 0.80, with standard deviations of 26.87 and 25.20.

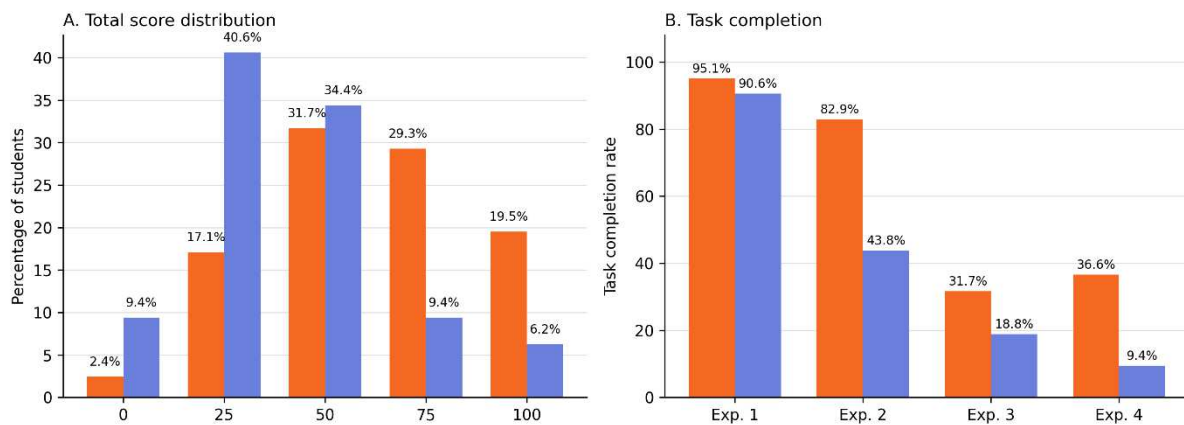


Figure 2. Stata assessment at the end of semester 2, by class

Note: The orange bars are class 4 and the blue bars are class 3. Class 4 received a further round of the bring-your-own-device arrangement; class 3 kept the standard schedule. Panel A plots the percentage of students in each class at each total score from 0 to 100. Panel B plots the percentage completing each of the four graded tasks.

Across both semesters the higher SPT classes pull ahead as tasks get harder. In Semester 1 the T4 gap sits at the bottom of the scale, where the control classes leave far more students stuck at one task or none. In semester 2 the two classes nearly tie on the easiest task, and the gap opens from the second task onward. The two semester 1 groups differed by under one point on a baseline that takes only two values. In semester 2, class 4 already stood 4.4 points ahead at T4, a prior gap that precedes the follow-up, so the added round cannot be read as its sole cause. The hardest tasks are the ones that demand a full AI-code workflow, and it is there that more SPT tracks better performance in both semesters.

3.3 Robustness of the Hands-On Estimate

Dropping any single class leaves the T4 group coefficient positive but moves its size. Across the four leave-one-class-out fits it runs from 7.79 to 13.68 around the main estimate of 10.81, lower when the stronger class in either condition is removed and higher when the weaker one is. Table 4 sets this alongside alternative covariate sets and a progress-score specification. The unadjusted difference is 11.17, conditioning on T1 gives 10.81, adding sex gives 10.88, and the T4-minus-T1 progress score gives 10.23. The sign holds in every fit, and each drop leaves one class facing two, so the four fits are sensitivity checks, not independent replications.

Table 4. Robustness of the hands-on group coefficient.

Block	Specification	Coef	N
Leave-one-class-out	Drop Class 1	7.79	103
	Drop Class 2	13.39	108
	Drop Class 3	8.50	106
	Drop Class 4	13.68	97
	Group only	11.17	138
Alternative covariate sets	+ T1	10.81	138
	+ T1 + Sex	10.88	138
Progress score, T4 – T1	$\Delta \sim \text{Group} + \text{Sex}$	10.23	138

Note: Each leave-one-class-out fit drops one class; with two classes per condition, each drop leaves one class facing two, so the four fits are sensitivity checks rather than independent replications.

4. Mechanism and the Remaining Bottleneck

4.1 How Supervised Practice Time Worked

The two conditions shared one instructor, one syllabus, one AI assistant, and the same teaching materials, and differed only in supervised practice time. With everything but supervised practice held constant, the question is what those hours did.

Figure 3 shows insufficient self-initiative as the top barrier to learning the AI-code workflow in both groups, at 63.8 percent in classes 1 and 2 and 69.6 percent in classes 3 and 4. Almost no one named the instructor, at 0 and 1.4 percent for the option that the teacher did not press them to practice. Lacking the self-initiative they report, students do most of their real practice in the classroom. Under the instructor's watch they run the prompt-run-inspect-revise cycle they would otherwise leave undone. This shift from watching to doing is what raises performance in Freeman et al. (2014). Class hours themselves raise attainment where they have been varied experimentally (Andersen et al. 2016), and

here the gap opens by T2, widens through T3, and holds at T4. We did not observe what students did while the machines were open or record practice outside class, so this reading links the questionnaires to the scores and does not measure the mechanism.

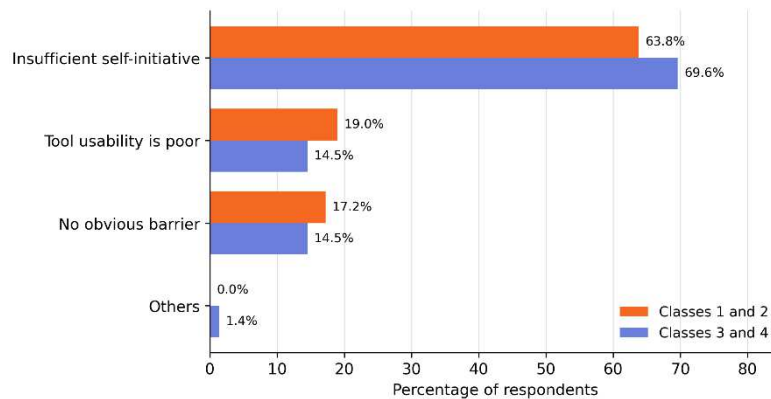


Figure 3. Most commonly reported barrier of learning AI-code workflow, by group.

Note: Bars show the percentage of respondents in each class group choosing each option for the item on the main barrier to learning the AI-code workflow.

This mechanism may reach other students and other software. The trait it turns on, a weak drive to practice without supervision, is unlikely to be local to this university. Autonomous learning motivation tends to decline across the undergraduate years (Pan & Gauvain, 2012), and applied learning experiences relate to stronger academic motivation (Trolan & Jach, 2020). The semester 2 Stata assessment is consistent with the workflow not being tied to a single package, though it rests on one class per condition.

4.2 The Unsolved Difficulty of Verifying AI Output

Once students run the AI-code workflow with ease, a second difficulty surfaces. Figure 4 places judging whether the output is correct first among the bottlenecks, at 51.7 percent in classes 1 and 2 and 46.4 percent in classes 3 and 4, ahead of writing prompts and reading errors. Both groups placed this bottleneck first. The two instruments were written by the same instructor and cover the same items in meaning, so this agreement reflects internal consistency across the two versions of one questionnaire.

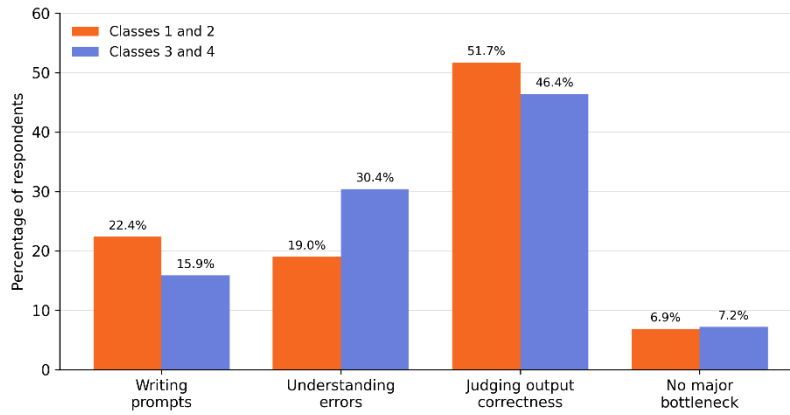


Figure 4. Most commonly reported bottleneck of using AI-code workflow, by group.

Note: Bars show the percentage of respondents in each class group choosing each option for the item on the main bottleneck in using AI-code workflow. Orange bars are classes 1 and 2 and blue bars are classes 3 and 4, after straight-lining screening.

Deep domain knowledge offers one route to verification, but an unlikely one for these students. Large language models can produce fluent output that is wrong in ways a reader without domain knowledge will not catch (Alansari & Luqman, 2026). Acquiring that knowledge asks for a long build-up after class that most of these students, low on self-initiative, will not sustain.

One direction worth exploring is to build verification inside the classroom instead. The SPT might carry a small set of quick checks that flag a wrong result without a full command of the underlying theory, a step the current SPT did not include. A compact, teachable routine is what such an attempt would aim at, and whether it closes the gap is left to future work.

4.3 Limitations

Several limitations bound what this study can claim. First, the design has few clusters. Four classes carry the first-semester contrast and two carry the follow-up, so it shows the size and direction of a difference but not its statistical significance, and we lean on effect sizes, distributions, and the leave-one-class-out fits reported in the robustness analysis. Class 4 compounds this, since it entered through the vocational channel on a different scoring scale and is not exchangeable with classes 1 to 3; dropping it moves the first-semester estimate little, but the follow-up rests on it alone, and its added hours cannot be told apart from its intake.

Second, the supervised-practice contrast is an upper bound, not a measured quantity. Attendance was recorded by session, but the split between demonstration and student practice within a session was not. We also did not record how many treatment students brought a working laptop to each session, and

nothing prevented control students from using DeepSeek on their own machines outside class. Each of these works against the contrast rather than for it. T1 is also not a strictly clean baseline: class 2 had one device session before sitting T1, while its paired control class had by then watched the same elementary demonstration of running supplied code, so the two entered T1 having met the workflow once, one by doing and one by watching.

Third, three narrower bounds remain. The hands-on scale is coarse, since four all-or-nothing tasks give five totals and make T1 a weak discriminator, and the instructor graded every script unblinded to condition. The questionnaires are descriptive and anonymous, so no response ties to a test score and cross-group readings hold only for identically worded items. The setting is also narrow, since one program, one instructor, and one AI assistant carry the study, and the two BYOD classes left the instructor's teaching after the first semester, leaving persistence unmeasured.

5. Conclusion

This study asked whether and how supervised practice time relates to finance students' command of the AI-code workflow. Our account has two parts. Across two semesters, more supervised hours went with a widening hands-on advantage, and a second-semester test in new software pointed the same way. The questionnaires point to a plainer account. Supervised hours move practice into the classroom, so more of the required rehearsal happens for students who report they would not do it alone.

What remains unsolved is verification. Students in both groups name judging whether the AI's answer is correct as their standing difficulty, once producing runnable code no longer binds. Accumulating the knowledge to judge well asks for after-class effort these students rarely give, so the workable direction points back to the supervised classroom. Whether a compact set of teachable checks can be built into those hours is the question this study leaves open.

References

- Alansari A, Luqman H (2026) Large language models hallucination: a comprehensive survey. *Comput Sci Rev* 61:100970. <https://doi.org/10.1016/j.cosrev.2026.100970>
- Andersen SC, Humlum MK, Nandrup AB (2016) Increasing instruction time in school does increase learning. *Proc Natl Acad Sci USA* 113:7481–7484. <https://doi.org/10.1073/pnas.1516686113>
- Angrist J, Lavy V (2002) New evidence on classroom computers and pupil learning. *Econ J* 112:735–765. <https://doi.org/10.1111/1468-0297.00068>

- Cepeda NJ, Pashler H, Vul E et al (2006) Distributed practice in verbal recall tasks: a review and quantitative synthesis. *Psychol Bull* 132:354–380. <https://doi.org/10.1037/0033-2909.132.3.354>
- Cheung ACK, Slavin RE (2013) The effectiveness of educational technology applications for enhancing mathematics achievement in K-12 classrooms: a meta-analysis. *Educ Res Rev* 9:88–113. <https://doi.org/10.1016/j.edurev.2013.01.001>
- Freeman S, Eddy SL, McDonough M et al (2014) Active learning increases student performance in science, engineering, and mathematics. *Proc Natl Acad Sci USA* 111:8410–8415. <https://doi.org/10.1073/pnas.1319030111>
- Kasneci E, Sessler K, Küchemann S et al (2023) ChatGPT for good? On opportunities and challenges of large language models for education. *Learn Individ Differ* 103:102274. <https://doi.org/10.1016/j.lindif.2023.102274>
- Kazemitabaar M, Chow J, Ma CKT et al (2023) Studying the effect of AI code generators on supporting novice learners in introductory programming. In: *Proceedings of the 2023 CHI conference on human factors in computing systems*, article 455. Association for Computing Machinery, New York. <https://doi.org/10.1145/3544548.3580919>
- Li B, Ning F, Zhang L et al (2021) Evaluation of a practice system supporting distributed practice for novice programming students. *J Pac Rim Psychol* 15:1–18. <https://doi.org/10.1177/18344909211008264>
- Pan Y, Gauvain M (2012) The continuity of college students' autonomous learning motivation and its predictors: a three-year longitudinal study. *Learn Individ Differ* 22:92–99. <https://doi.org/10.1016/j.lindif.2011.11.010>
- Prather J, Denny P, Leinonen J et al (2024) The robots are here: navigating the generative AI revolution in computing education. In: *Proceedings of the 2023 working group reports on innovation and technology in computer science education*. Association for Computing Machinery, New York, p 108–159. <https://doi.org/10.1145/3623762.3633499>
- Sung Y-T, Chang K-E, Liu T-C (2016) The effects of integrating mobile devices with teaching and learning on students' learning performance: a meta-analysis and research synthesis. *Computers & Education*. 94:252–275. <https://doi.org/10.1016/j.compedu.2015.11.008>
- Theobald EJ, Hill MJ, Tran E et al (2020) Active learning narrows achievement gaps for underrepresented students in undergraduate science, technology, engineering, and math. *Proc Natl Acad Sci USA* 117:6476–6483. <https://doi.org/10.1073/pnas.1916903117>

Trolian TL, Jach EA (2020) Engagement in college and university applied learning experiences and students' academic motivation. *Journal of Experiential Education* 43:317–335.
<https://doi.org/10.1177/1053825920925100>

Zheng B, Warschauer M, Lin C-H, Chang C (2016) Learning in one-to-one laptop environments: a meta-analysis and research synthesis. *Review of Educational Research*. 86:1052–1084.
<https://doi.org/10.3102/0034654316628645>

Appendix

Table A1. Baseline characteristics by condition

Group and class	n	Female, %	T1 mean (SD)
Treatment	65	55.4	8.46 (11.92)
Class 1	35	51.4	7.86 (11.78)
Class 2	30	60.0	9.17 (12.25)
Control	73	50.7	7.53 (11.55)
Class 3	32	43.8	8.59 (12.06)
Class 4	41	56.1	6.71 (11.21)

Note: T1 is the first hands-on assessment, taken in weeks 1 to 2. Sex and the T1 score are the only baseline variables available, since the program records no measure of prior programming or prior AI use, students reported no prior programming at entry, and class 4's admission composite is not on the same scale as the general admission score. Every T1 score was 0 or 25, so T1 separates students who completed one task from those who completed none.

Table A2. Item-level response distributions for the MATLAB course follow-up questionnaire (reporting labels Class 1 and Class 2)

Item	Stem and response options	Analyzed n	Analyzed %
A2.1	Class		
	Class 1	32	55.2
	Class 2	26	44.8
A2.2	When you meet a finance problem that requires writing code, do you use the AI-code workflow you learned in the MATLAB course?		
	I use it on my own initiative	44	75.9
	I do not think of using it unless someone reminds me	12	20.7
	I no longer know how to use it	2	3.4
A2.3	Without an instructor guiding you, how confident are you that you could use AI-code workflow to solve a complex financial computation problem on your own?		
	Confident that I could complete it independently	7	12.1
	Could generally do it, with occasional sticking points	35	60.3

	Could do it only with someone's guidance	15	25.9
	Essentially could not do it	1	1.7
A2.4	When AI-generated code returns an error, how much of it can you now resolve yourself?		
	Most errors	15	25.9
	About half	35	60.3
	Rarely any	8	13.8
A2.5	Compared with the point at which the MATLAB course ended, your present fluency in using AI-code workflow on finance problems is		
	More fluent	28	48.3
	About the same	18	31.0
	Less fluent than before	12	20.7
A2.6	If your corporate finance course also included AI-code workflow practice, would you want it?		
	Yes	28	48.3
	No	3	5.2
	Indifferent	3	5.2
	Yes, provided that it does not put passing the final examination at risk	24	41.4
A2.7	To which kind of course would you like AI-code workflow teaching to be extended?		
	More theoretical courses, such as finance theory	6	10.3
	More applied courses, such as investments or financial statement analysis	37	63.8
	Both	15	25.9
	I would not like it extended	0	0.0
A2.8	If the length of a class session is fixed, where would you prefer the instructor to spend more of it?		
	Instructor demonstration and explanation	7	12.1
	Students working hands-on	6	10.3
	Half on each	45	77.6
A2.9	Which ability would you rather an AI-code workflow practice course develop?		
	Producing results quickly with AI	14	24.1
	Judging whether AI output is correct	3	5.2
	Both	41	70.7
A2.10	When you use AI-code workflow on a finance task, at which step do you most often get stuck?		
	Not knowing how to write the prompt	13	22.4
	Not being able to read error messages	11	19.0
	Not being able to judge whether the AI output is correct	30	51.7
	Rarely get stuck	4	6.9
A2.11	In your learning of AI-code workflow, the greatest obstacle is		
	My own initiative is insufficient	37	63.8
	The tools themselves are hard to use	11	19.0

	The instructor did not press us to practice	0	0.0
	No obvious obstacle	10	17.2
A2.12	Overall, the share of a corporate finance course that should be devoted to AI-code workflow practice is		
	100%	5	8.6
	50%	43	74.1
	No more than 30%	10	17.2
A2.13	Would you recommend this AI-code workflow practice format to students in the years below you?		
	Yes	53	91.4
	No	5	8.6

Notes. All 13 items were single-choice, administered to Class 1 and Class 2 approximately six months after the MATLAB course ended. Of 65 enrolled students, 59 responded, 90.8%; one response was excluded for straight-lining, leaving 58 analyzed responses. Unselected options are retained and reported as 0 (0.0).

Table A3. Item-level response distributions for the corporate finance course questionnaire (reporting labels Class 3 and Class 4)

Item	Stem and response options	Analyzed n	Analyzed %
A3.1	Class		
	Class 4	40	58.0
	Class 3	29	42.0
A3.2	If you were now asked to complete a financial computation task with AI on your own, with no instructor present, how confident are you?		
	Confident that I could complete it independently	10	14.5
	Could generally do it, with occasional sticking points	45	65.2
	Could do it only with someone's guidance	12	17.4
	Essentially could not do it	2	2.9
A3.3	When AI-generated code returns an error, how much of it can you now resolve yourself?		
	Most errors	21	30.4
	About half	39	56.5
	Rarely any	9	13.0
A3.4	Compared with the point at which the MATLAB course ended, your present fluency in using AI on finance problems is		
	More fluent	47	68.1
	About the same	17	24.6
	Less fluent than before	5	7.2
A3.5	How helpful was the AI-code workflow learned in the MATLAB course for the corporate finance course?		
	Very helpful	35	50.7
	Somewhat helpful	33	47.8
	Almost no help	1	1.4

A3.6	Beyond corporate finance, to which kind of course would you like AI-code workflow teaching to be extended?		
	More theoretical courses, such as finance theory	7	10.1
	More applied courses, such as investments or financial statement analysis	33	47.8
	Both	26	37.7
	I would not like it extended	3	4.3
A3.7	If the length of a class session is fixed, where would you prefer the instructor to spend more of it?		
	Instructor demonstration and explanation	12	17.4
	Students working hands-on	14	20.3
	Half on each	43	62.3
A3.8	Which classroom arrangement do you feel gives you the most solid grasp of the material?		
	Working through the material in step with the instructor	35	50.7
	Instructor lectures first, students work on their own afterwards	19	27.5
	Working directly on an authentic project and asking for help when stuck	15	21.7
A3.9	Which ability would you rather an AI-code workflow course develop?		
	Producing results quickly with AI	11	15.9
	Judging whether AI output is correct	6	8.7
	Both	52	75.4
A3.10	On the question of discontinuing or retaining AI-code workflow practice in corporate finance, your position is		
	It should be retained	52	75.4
	It should be discontinued	4	5.8
	Indifferent	13	18.8
A3.11	When you use AI-code workflow on a finance task, at which step do you most often get stuck?		
	Not knowing how to write the prompt	11	15.9
	Not being able to read error messages	21	30.4
	Not being able to judge whether the AI output is correct	32	46.4
	Rarely get stuck	5	7.2
A3.12	In your learning of AI-code workflow, the greatest obstacle is		
	My own initiative is insufficient	48	69.6
	The tools themselves are hard to use	10	14.5
	The instructor did not press us to practice	1	1.4
	No obvious obstacle	10	14.5
A3.13	Overall, how satisfied are you with the use of AI-code workflow teaching in this course?		
	Very satisfied	21	30.4
	Satisfied	42	60.9
	Neutral	5	7.2
	Dissatisfied	1	1.4

A3.14	Would you recommend this AI-code workflow practice format to students in the years below you?		
	Yes	63	91.3
	No	6	8.7
A3.15	Had you used Stata before taking the corporate finance course?		
	Yes	11	15.9
	No	58	84.1
A3.15	Item A3.15 by class, respondents reporting prior use of Stata		
	Class 4	5 of 40	12.5
	Class 3	6 of 29	20.7

Notes. All 15 items were single-choice, administered to Class 3 and Class 4 at the end of the corporate finance course. Of 73 enrolled students, 72 responded, 98.6%; three responses, all from Class 3, were excluded for straight-lining, leaving 69 analyzed responses. Unselected options are retained and reported as 0.

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

AI Statement

During the preparation of this manuscript the authors used a generative AI tool for language editing only. No content was generated by AI. The authors reviewed and take responsibility for the final text.

Data Availability

The anonymized assessment data, questionnaire data, questionnaire summaries, variable documentation, and reproducibility code supporting this study are available in the supplementary package titled “Supplementary Materials for Peer Review” submitted with this revision.

Ethics Statement

This study was reviewed by the relevant institutional research administration department. The department determined that the research posed no more than the risks encountered in routine educational activities and granted an exemption from full ethics review and a waiver of written informed consent. The research was conducted in accordance with applicable institutional research ethics requirements and Chinese regulations concerning research ethics and personal information protection. The data were anonymized before analysis, access to the research data was restricted, and the findings are reported

only in aggregate form. The institutional name and approval number have been withheld to preserve double-blind review.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SupplementaryMaterialsforPeerReview.zip](#)