ADVAIT SARKAR

HOW TO DEFINE
AN ELEPHANT

First edition, March 2017

Contents

1

Introduction

Like an elephant, a supervision is easy to recognise but difficult to define.
– Frank King, Notes on supervising Computer Science, 1995

The form of a supervision at the University of Cambridge is suspiciously simple: a supervisor assigns some work to be completed; the students, typically two or three, complete the work. Then, at a predetermined time and place, students and supervisor meet. They talk for an hour, perhaps some things are written down, and then everyone leaves. This innocuous performance, so commonplace that it occurs hundreds, if not thousands of times every day during Cambridge terms, surely cannot sustain much further deliberation?

Despite its simplicity, the Cambridge supervision system is something we jealously pride ourselves on, the last inconquerable bastion of the 'old guard' of higher education, unfazed by open online courses and other heretic innovations, the irreplaceable, insurmountable learning experience which produces generation after generation of the world's finest minds.

And yet, we do little by way of preserving the *spirit* of supervisions. What really is the *purpose* of a supervision? What makes a supervision good? What makes it successful? The answers to these questions lie in the soft collective knowledge of thousands of supervisors scattered across the university. As new supervisors arrive, they are expected to assimilate these values by virtue of osmosis (another of the many ways in which Cambridge is still very visibly a product of its monastic origins). Many of these new supervisors ultimately 'get it', in turn becoming responsible for curating the supervision as a lived experience. In fact, the curation, constant experimentation, and reinvention of the supervision is one of its hallmarks; attempting to regulate or clearly define a format for it would defeat its purpose. It is a wonderfully self-sustaining system, and of course, no written document can ever capture it in its entirety. But I feel that I am in a position to contribute a document which, while certainly not a replacement for this assimilation

process, can be a facilitator for new supervisors, and reduce the fragility inherent in a system which depends largely on oral and experiential tradition. Long-time observers may not feel that the system is fragile, but the reality is that the mantle of supervision is increasingly being assumed by graduate students, research assistants, and postdoctoral researchers, who have neither the time nor the academic network to be able to observe the supervision through a critical lens.

Thus, my aim here is not to reproduce basic information about the logistics of supervising, such as how to register yourself to supervise, how many supervisions to give, etc., as such information is easily found elsewhere. Nor have I excluded all such information – it is impossible to adequately treat certain aspects of supervision theory without discussing logistics.[1] Rather, my primary objective is to explicate my personal theory of how to create value in supervisions, to codify the thoughts and experience I have developed over the course of my nearly 8 years at Cambridge, during which I have participated in over 300 hours of supervisions (as supervisee and supervisor in roughly equal proportions). At the time of this writing, I have just completed my PhD, and in the coming months, I will conduct my last few supervisions for the foreseeable future. As such, I feel now is a good time to attempt to preserve some of what I have learnt, before my memories of supervisions become clouded, rose-tinted, and otherwise muddled.

For whom is this book? Perhaps you are new to Cambridge. You are a new PhD student, postdoc, or lecturer. You are intrigued by this mysterious supervision system and want to get involved – this book is for you. Perhaps you are a highly experienced supervisor and wish to see whether what you think you know about supervision is shared by another experienced supervisor – this book is for you. Perhaps you are a Cambridge student, trying to make sense of how to get value out of supervisions – this book could even be for you.

Is this a guide to supervising? It might look like it, and in some places might come across as prescriptive – for this I apologise. No, it is not a guide to supervising. It is a summary, to the best of my abilities, of the attitudes, theories, and practices that *I* have personally found to be valuable for *my* supervisions.

Then what will the reader learn? The advice in this book (like my research, unsurprisingly) is *provisional, contingent, and aspirational*, in the words of the great Bill Gaver. Take what you will from it. I sincerely hope you are able to identify aspects of my practice that can be applied in your own context, and that by observing how my own thoughts on supervising have developed, you can critically reflect on your own supervision practice.

[1] For example, chapter 5 contains a fairly lengthy discussion on whether to use paper or a whiteboard during a supervision. Perhaps a relatively mundane logistical concern, but amongst other things, it has implications for managing the perception of power structure in supervisions.

I have tried to capture my objective in the title of this book; Dr King's pithy observation cuts straight to the heart of the problem, which is that *no one knows what the definition of a supervision is*. No one ever will, and that is an essential virtue of the supervision. But maybe, just maybe, we can learn *how* to define the elephant, which is perhaps more important than ultimately arriving at a definition.

Unquestionably, there are people at Cambridge with vastly more experience and knowledge than me, with even more highly nuanced critical and pedagogic lenses. However, being in possession of these refined faculties, they have far better work to apply themselves to than writing things like this. For better or worse, dear reader, we are stuck in this dance together.

2

The supervision

What is the purpose of the supervision? *There is no answer* to this question, but I shall attempt to supply one regardless. Over the years, I have come to the conclusion that framing the objective of the supervision as follows is not only effective for directing the effort of supervisors, but also is the optimal use of the opportunities inherent in the supervision system.

The supervision provides structure and individualised feedback for high quality self-study.

Say it with me:

The supervision provides structure and individualised feedback for high quality self-study.

The supervision provides structure and individualised feedback for high quality self-study.

The supervision provides structure and individualised feedback for high quality self-study.

Let me unpack each of the terms in this definition:

THE SUPERVISION: Why 'the' supervision, and not 'a' supervision? I am talking, from the students' perspective, about the entire experience of supervisions as a complementary part of their education, rather than the actual hour of 'a' supervision itself. 'A' supervision is indubitably an important part of 'the' supervision of a student, but 'the' supervision is broader, and encompasses activities that take place before and after 'a' supervision.

HIGH QUALITY SELF-STUDY: This is perhaps the most important, but least intuitive part of the definition. The aim of the supervision is primarily to create an environment where the student experiences a high quality of self study. "That's bizarre!" I hear you exclaim, "surely some teaching is meant to happen during a supervision!" Yes, indeed. But I'm not talking about just a supervision,

I'm talking about the 3-4 hours of work the student puts in before the supervision even happens. Ensuring the quality of the learning experience that occurs during that work is the single most important responsibility of the supervisor.

STRUCTURE: The student has received new information at the lecture, and the purpose of self-study is to assimilate that information, to contextualise it, to relate the minutiae – the equations, the algorithms, the papers, the assigned reading – to the broader picture of human needs and academic discourse. In other words, to turn that information into knowledge. This they cannot do by themselves without guidance; merely revising their own notes and lecture handouts will not achieve this. Self-study at the undergraduate level requires at least a modicum of structure. There are many ways for the supervisor to structure self-study, the most important of which is the setting of appropriate work, and the creation of tasks that force them to work through the implications of the information delivered in the lectures and create a deeper understanding from it. During the supervision itself, the work done during the self-study can be further contextualised, which can help students structure their self-study for further supervisions.

INDIVIDUALISED FEEDBACK: This is the killer feature of the supervision. Lectures and interesting assignments can be replaced by engaging and interactive video materials, and indeed this is the approach of open online courses. Universities and schools around the world are experimenting with the 'flipped classroom', wherein students first imbibe information through reading and video materials, and then the actual allocated lecture time is spent discussing, interrogating, reflecting, contextualising, and assimilating this information, facilitated by the lecturer. Luckily for us, the Cambridge supervision system has that built in by default. And moreover, the University has built a (perhaps inefficient, but at least functional) system for students to have flipped classroom experiences in a class of only two or three students. It is truly a remarkable feat to organise the tens of thousands of supervisions that occur each year across the university, given how demanding supervisions are in terms of time and expertise. A large format flipped classroom cannot give every student individual time to express their thoughts and interrogate their understanding, but a supervision of two or three students can – and this is the core, unique strength of Cambridge (and Oxford) that can never be replaced using an online course.

Supervisions for teaching thought

An alternative perspective is that it is not so much the high quality self-study that is important, but teaching students *how to think* that is the key outcome. That is, the aim of supervisions is to show students that what they are learning is a discipline, not the absolute Truth, but rather a way of thinking with (possibly flawed) practices and methods, that have evolved over time and are a product of history, influential thinkers, and socio-cultural context. The subject matter they learn, inevitably, will become outdated. In this view, the aim of supervisions is to equip students with the thinking tools required to learn new concepts, ideologies, even entire disciplines by themselves. To this I say: Yes! Of course! We must send our graduates into the world with a critical appreciation of their discipline, thoroughly empowered to direct their own lifelong learning. Nonetheless, there are a few reasons to believe that my framing of supervisions as creating the opportunity for high quality self-study is even better:

- First, proponents of this view must concede that discussions of ways of thinking must necessarily be grounded, for the benefit of the undergraduates, in the actual subject matter they are studying.[1] In order for students to be able to transcend the subject matter and engage in a meta-discourse about the disciplinary nature of that matter, they must first achieve mastery of the subject matter! And the best way to achieve mastery of subject matter is through high quality self-study.

- Second, high quality self-study is necessary, and sometimes sufficient, to create an awareness of the ways of thinking. Therefore, expressing the objective of the supervision in terms of self-study does not preclude supervisions as teaching thought. Only through the careful setting of provocative and reflective work, can you lay the groundwork for students to be receptive to thinking about thinking – you may wax on about history and context during a supervision all you like, but if the students have not been put in the correct frame of mind you will likely do more damage than good.

- Third, the high quality self-study definition is more immediately practicable. Telling a supervisor that they should teach students 'how to think' is more a platitude than an aphorism. Moreover, teaching thinking effectively is an individualised skill that supervisors develop with time, and involves technique that is well beyond what can be transferred through written or spoken guidelines.[2] In contrast, as I will outline over the next three chapters, considering the self-study experience of students gives rise to many helpful practical suggestions for supervisors.

[1] If you refuse to concede this, then you are essentially suggesting that all undergraduates must study only philosophy, which is an interesting proposition, but hideously impractical in the 21st century economy. If you further object that the purpose of University is not to accede to economic pressures and serve short-term societal needs (after all, that is what vocational institutes are for) then I would agree with you up to a point but it would take rather more than a footnote to flesh out that line of reasoning. But in brief: the conclusion is essentially either (a) that at some point one must draw an arbitrary line that delineates market value from perceived future value, or (b) nihilist: that Universities are fit for teaching nothing at all. Pragmatism, however, is in my favour.

[2] Essentially, I am saying that even though it is desirable for supervisors to teach students how to think, you cannot simply *teach supervisors* to teach students how to think – it is tacit knowledge.

Don't panic

This may all seem quite daunting, and appear to be a lot of responsibility, especially for new supervisors. This is only fair, as it is indeed a big responsibility. However, supervisors may take some comfort in the fact that if they have set high quality work that has been completed by the students to a high standard, then the majority of the purpose of the supervision has already been fulfilled. A lot of energy is spent thinking about how the hour of the supervision will proceed, and that hour is indeed important, but in reality that hour contributes only a part of the true value of supervisions. By the time the supervision actually happens, a lot of action has already taken place, when the students engaged with the assigned work.

3

The student

The supervision cannot happen in abstract isolation from the rest of the students' experience. The supervision fits into a broad context of the student's overall academic, social, and intellectual trajectory. In order to facilitate the best learning experience, one must consider various such aspects. This moves us beyond the concept merely of the student, to build up a picture of the *student-in-context*.

The lecture-supervision dynamic

Supervisions are delivered in conjunction with a specific course of lectures. Correspondingly, the progression of that lecture course is an important aspect of the context. Supervisions must take care to be deeply connected to, and in service of, the content of the lectures. The best way to understand how your supervisions must complement the lectures as delivered is to attend the lectures yourself. This gives you a sense for how the lecturer themself sets out the context and objectives of the delivered content, and what parts of the material the lecturer emphasises or de-emphasises, which may differ from what the lecture notes and handouts would imply.

Timing your supervisions with lectures takes a little effort, but is well worth the improvement in coherence of supervisions. In computer science, the recommendation is to have one supervision for every four lectures, and the supervisions almost always run concurrently[1] with the lectures. This means that for a course of 16 lectures, you are likely going to plan 4 supervisions, and begin delivering them as soon as students have covered enough material to tackle their first set of supervision assignments. Ideally, the first supervision will cover material from the first subset of four lectures, the next supervision from the next subset of four, and so on. Ideally, you will allow adequate time after the fourth lecture in a subset for the students to do the assigned work and submit it before the supervision. It is not always possible to map supervision tasks this neatly, especially given how complex it can be to

[1] In the sense that the lecture course might run for 4-8 weeks, with 2-3 lectures a week, and most supervisions for that course should be scheduled for those same weeks.

timetable supervisions. However, it is usually far better to postpone a supervision to allow time for all necessary lectures to have been attended, than it is to require students to read ahead in order to answer supervision questions.

It is not always possible to attend the lectures yourself. In this case it is imperative that you check whether the lectures are proceeding according to the structure implied by the course notes. This is best done by asking the students themselves. Sometimes lecturers will switch the order of lectures, sometimes they will be running behind, or (occasionally) running faster than anticipated. This will mess up your supervision plans if a lecture you were counting on to have been delivered by a certain date has not been delivered. Asking students to keep you notified of any such aberrations will help you anticipate and avert crises, such as students showing up at the supervision with only half the assigned work completed because they are yet to have the remainder of the relevant lectures.

A most effective strategy is to notify students of *all* the assigned work for *all* supervisions at the beginning of the course. This is useful for you, since this entails planning through your entire series of supervisions beforehand, which usually results in a more coherent set of assignments. You will also be relieved of the bother of disseminating the assigned work every week. More importantly, it is immensely useful for the students, because by keeping the supervision assignment in mind during lectures, they can actively start to formulate their approaches and answers during the lectures themselves. They can use their time more efficiently, by completing parts of your assigned work as and when they have received coverage of the relevant issue in lectures, and it is never the case that they have cleared time to work but are unable to do so because they don't know what the assignment is. If during the course of the supervisions, you identify other assignments or readings that would be particularly useful for a specific student or specific groups, then it is completely acceptable to send out amendments to the required work (as long as you have given sufficient notice).

Relationship to other courses

The courses of the tripos are designed with great care so as to create a holistic undergraduate education.[2] As such, they are not courses that exist in isolation, with neatly delineated boundaries, but rather, they feed into each other. In the most obvious sense, some courses are prerequisites for other courses. It would, for instance, be difficult to study complexity theory at the level of abstraction and formality taught in the second year, without the initial introduction to computational complexity in the algorithms course taught in the first year. Similarly, third year signal pro-

[2] One may argue whether they actually achieve this, and how they may be improved, but that is out of scope.

cessing cannot be taught without the prerequisite introduction to Fourier methods taught in the second year.[3] But there is a second, deeper sense of interrelatedness between courses that, unless made explicit, may pass by the undergraduates unnoticed. For instance, the relationship between natural language processing and information retrieval. Or the relationship between information retrieval and machine learning. Or the relationship between machine learning and human-computer interaction. These are relationships not only in content – the methods from each of these topics have interesting and useful applications in each of the others – but also in terms of disciplinary history; i.e, how these disciplines have come to be in their current state.

A skilled supervisor understands how the course being supervised fits into the other courses in the tripos in both senses. A simple consequence of this awareness is that a good signal processing supervisor will not presuppose a more nuanced understanding of Fourier methods than what the students have previously studied. Conversely, a good Fourier methods supervisor will understand that they are providing the relevant basis for study in future years, and will structure their teaching accordingly – perhaps even by 'priming' students for their following years by introducing simple signal processing examples in their assignments. Moreover, a great natural language processing supervisor will draw students' attention to the links between natural language processing, information retrieval, machine learning, and human-computer interaction.

Year, term, week

Where your students are within their Cambridge degree timeline affects their attitudes, expected outcomes, and investment in supervisions. Their timeline can be effectively divided into three dimensions: the year, the term, and the week.

YEAR: The most visible effect of the year in which your students are is how well they are able to take advantage of supervisions. Final year students (third years, for most degrees) have typically already been in 100 or more supervisions, so they are more demanding and engaged. They have fewer inhibitions during supervisions, are more likely to suggest new material and alternative topics for discussion during the supervision, and are more likely to seek out the 'bigger picture' of the material they are learning. They (hopefully) by now have developed a tacit awareness that self-study is the most important aspect of the supervision experience. They are also acutely aware of the importance of final year exams, and so are typically more heavily invested in exam preparation. Final year students are generally under the most pressure, having to engage with their dissertation alongside supervision work, exam preparation,

[3] Readers from other disciplines are welcome to substitute their own course names as appropriate here. Additionally – the computer science tripos itself is subject to frequent change, so these courses may well have different names and be introduced at different years in the future. I am simply expressing the idea that understanding some concepts is a prerequisite to understanding others (an idea which itself can be debated and further qualified, but that is out of scope) and this is often reflected in the design of tripos courses.

and applications for jobs and graduate programs. These are things to bear in mind when supervising final year undergraduates.

In contrast, first years can be heavily inhibited during supervisions, afraid of saying the 'wrong' thing or appearing unintelligent. They are still developing their understanding of supervision etiquette and turn-taking with their supervision partners. They may misunderstand the balance between self-study and supervision time. They are still forming their view of supervisions and the role they play in their education. First years are also inexorably preoccupied with the novelties of university life. However, first years are most open to supervision styles that do not directly assist them in their exam preparations. The challenge with first years is to swiftly impress upon them that the supervision is *their* time to get individualised feedback on their structured, high-quality self study, that they should participate actively in supervisions, that they should see supervision partners as an opportunity for peer learning and teaching (as opposed to a competitive threat), and that you are here to help them, not judge them.

TERM: The gradient of exam-orientation that starts slowly in first year and reaches fever pitch in final year also has a parallel in the Cambridge term. Regardless of whether students are in first, second, or third year, they are least exam-oriented in Michaelmas term, and most exam-oriented in Easter term.[4] As will shortly be outlined, exam preparation is only one of multiple potential outcomes that students expect from supervisions, but it is the outcome they are likely to be most vocal about in Easter term. If you wish to experiment with alternative supervision activities that involve extensive reading outside of the curriculum, or deep-diving into a relatively small but interesting part of the curriculum, students are most likely to be welcoming of this during Michaelmas term. Similarly, incorporating tripos question practice as part of Easter term supervisions is likely to be very well appreciated by exam-crazed students.

WEEK: Finally, consider the week of term in which you deliver your supervisions. You might have heard of the infamous Cambridge 'week 5 blues', referring to the phenomenon of students being overburdened with supervision work around the middle of term. Correspondingly, students may appear more stressed and flustered for supervisions you conduct in weeks 5-7. At this time, a decline in the quality of supervision work is not uncommon, and is usually not indicative of a drop in student performance. If in any doubt, ask the student whether the decline in quality was attributable to the week 5 focus on producing answers to satisfy the minimum requirements of the supervision assignment, or whether they are experiencing genuine difficulties with the material. One useful strategy is to plan some supervisions at the start of Lent or Easter term, for courses that ended in Michaelmas or Lent respectively.

[4] So heavy is the emphasis on exam preparation during Easter term that, unsurprisingly, it is often referred to as Exam term.

Supervisions rarely take place in the first two weeks of term, and students might have more time to attend to your assignments. The downside of this for the students is that the supervision is somewhat far removed from the lectures, and it can be hard to recall the lecture material when attempting the supervision work. However, you musn't feel too guilty, as they will have to revise this material for their exams anyway.

Individual academic trajectory

It is difficult in computer science to get a sense of how the student is developing over time, and how to assess the student within their own appropriate frame of reference. It is difficult because supervision is on a per-module basis, and so supervisors may only meet students for between 2 and 4 supervisions, on average.

Due to the variety of courses and students that I have supervised over the last few years, there have been some students who I have supervised throughout their undergraduate career, for courses in their first, second and third years. Something that struck me is that this additional context really helped how I interpreted student work and wrote supervision reports. For example, I supervised two students, Alice and Bob,[5] for multiple courses throughout their undergraduate years. In their third year, Alice and Bob both handed in slightly mediocre but acceptable written work for a supervision. Now, Alice had always struck me as a very bright student, she had always handed in excellent written work and engaged very well in the supervision, so for her to hand in work that was merely acceptable sent up warning flags. Conversely, Bob had struggled throughout his undergraduate career – often not handing in any work, missing supervisions, and generally not engaging with the material – so getting a decent piece of work from him was a good sign that he was beginning to improve. Consequently, the written supervision reports were quite different for these two individuals. However, had I no sense of the academic context of Alice and Bob, I would have written almost identical reports. This lack of context is, unfortunately, so common in computer science supervision. I suspect a great many of the supervision reports I have written would have been significantly different if I had a better idea of the student's prior history. In certain other subjects, first year Physics, for instance, it is typical to have a single supervisor teach all modules through the year, which helps them develop a better sense of each student's individual trajectory.

[5] Not real names or genders, obviously.

So what can you do if you are not aware of the student's academic history? The answer is: not much, unfortunately. You can try asking for a summary from the director of studies, but there is no guarantee that they will be able to provide any description of satisfactory

detail or utility. At most, they may alert you to students experiencing difficulties that may impact their work, but you are unlikely to be able to paint a picture as nuanced as the one that enabled me to write the reports for Alice and Bob. Perhaps at some point in the future it will be possible for supervisors to view reports written by other supervisors (if the student and supervisors consent to such information being shared). Until then, when you write your supervision reports:

- appreciate that you may not have the full picture, and

- be aware that the director of studies is one of the few people to oversee a student's entire undergraduate development, and your reports are a vital tool for them to build up that picture.

Expected outcomes and motivations

I have already alluded to the idea that exam preparation is one outcome that students expect from supervisions. What other outcomes might they expect? I have, broadly, observed at least four different outcomes for supervisions that are appropriate to try and address in the supervision: Skills, Interest, Knowledge, and Exams. These can be conveniently remembered as the acronym 'SIKE'.

- **Skills**: skills-motivated students want to know how what they are learning can be applied to real-world problems. That is, they want to see how this knowledge can be turned into a skill. Such students enjoy having discussions of real world cases where the algorithm/technique/etc. is used. They enjoy discussing what happens when you take a pure, abstract idea, and have to make it perform in the dirty environment of the real world, where there is finite memory, finite computation, finite data, and other real-world constraints. They enjoy hands-on, applied supervision work that asks them to write code and build solutions.

  Often, these students will have relevant industry experience from internships, or from their second year group project / final year project, that they can bring to bear on the discussion. Use this as an opportunity to engage them more deeply in the supervision and provide peer instruction.

- **Interest**: another outcome for supervisions is simply to get students more interested in the topic. Some students may not see it as being of any value to their own career aspirations, and are looking for a reason to invest themselves into learning it. Many strategies can be used to try and foster interest in these students. Often, these students will enjoy writing open-ended answers to larger problems posed by the topic, rather than an assignment

that requires getting bogged down in the minutiae of a hard technical derivation. They may also respond to practical examples, wider reading including contemporary issues and popular media, and whimsically-worded problems.

- **Knowledge**: knowledge-motivated students are interested in the topic and want to learn more. They are intrigued by, and perhaps critical of, the story presented in the lectures and want to dig deeper. They want more context, perhaps historical context. Such students enjoy wider reading. They enjoy tasks that expose them to alternative algorithms, solutions, protocols, and notations than those presented in the lectures, and ask to critically compare them. They may enjoy a deep-dive into an aspect of the topic that was merely glossed over in the lectures, to give them a greater sense of mastery over the topic. These students may potentially make good research students in the future.

- **Exams**: as already mentioned, exam-motivated students want to use the supervision as a revision technique. These students are easily appeased using a broad set of tripos questions as the assignment. For these students, discussion during the actual supervision would ideally consist of detailed feedback on their supervision work, exam strategies, tips for tackling different types of questions, and exercises such as guessing the implicit marking scheme for a question.[6]

[6] For example, a six mark question that states 'Define three types of fooglefargs' likely awards a single mark for correctly recalling a type of fooglefarg, and a further mark for defining it.

Of course, students are not motivated exclusively by one or the other of the above categories. Rather, they are motivated by differently weighted combinations of each, subject to fluctuation between years, terms, and courses. It is important for you as a supervisor to try and balance how you cater to each of these outcomes, with respect to both what you perceive the student is motivated by, as well as what you personally wish to impart to the students.

A conspicuous omission from the four outcomes above is 'learning'. What about 'learning'? Should students come to signal processing supervisions to learn signal processing? Should they come to mathematical methods supervisions to learn mathematical methods? It would certainly allow me to take the harsh acronym SIKE and turn it into the more mellifluous LIKES. Indeed, some students may come to the supervision expecting to *learn* the topic. However, this is a great opportunity to hammer home the point that the supervision is not where primary learning takes place. That is is best done in lectures and self-study. Of course, you want your students to learn *something* during the supervision, but not the primary subject material delivered in the lectures. If you find yourself recapitulating primary material during the supervision, something has gone wrong – either the lecture was unintelligible, or the student has not put in adequate work, or the student is having other difficulties.

Demographic factors

A final set of important considerations are those of the demographic factors of the student. These include gender, race, socioeconomic background, educational background, neurological typicality, and general ability level. Particularly in the first year, when students are still learning to loosen their inhibitions and participate in appropriate supervision turn-taking etiquette, you must take care to ensure that students do not feel disadvantaged or inhibited for these types of reasons. It is important that everyone feels equally entitled to participate in the supervision and engages equally well. This is one area where you really must constantly apply your professional judgment and moderate the supervision as a benevolent authority figure. If you feel for any reason that there are factors beyond your control that are inhibiting the performance of the student, then you must notify the director of studies immediately.

4

The assignment

The quality of the supervision assignments you set determines the opportunity for the students' self-study to have high quality. Designing supervisions requires a consideration of all the factors outlined in the previous chapter. Try to be consciously aware of the balance of Skills, Interest, Knowledge and Exam practice you are creating.

Quantity

How much work should you set? This varies greatly by discipline. In computer science, the rule of thumb is that students are expected to spend between 3 and 4 hours on work for each supervision. In constructing your set assignments, you must engage in the activity of viewing the work through the lens of the student, and estimating how much time it might take them to complete. This is not as easy as it sounds, and it is better to err on the side of giving students less work than more. It's important to constantly calibrate your expectations and assumptions, which is best done by directly asking students how they felt about the time investment required to do the work for a supervision; was it too much, too little, or just right?

Be aware of the year/term/week context. Since students are in general busier during weeks 5-7, their perceptions of workload at that time might be skewed. You might wish to plan your supervision work so that any particularly demanding assignments are tackled outside of this stressful period. For instance, you might notice that for a given course, there is one particular concept that is routinely challenging for all students. You might have discovered that making students trudge through a long and tiresome sequence of logic and derivation is usually effective at helping students get to grips with this concept. You know that in the past, some students have taken 5, or even 6 hours to do this assignment. If possible, try and plan your supervision work so that students don't have to slay this minotaur of an assignment during week 5 or during exam term.[1]

[1] If the course you're supervising happens to take place in exam term, though, you're out of luck.

A technique that some supervisors apply, which you might find to be of use, is to provide a core 'compulsory' set of assignments or questions to answer, but then include a number of additional optional questions in case the student feels that the core assignments were completed too quickly or easily. Optional further assignments are popular with knowledge-motivated students, but can be tricky to handle during the supervision, since not all students will have completed them.

Types of assignments

While many humanities subjects are almost entirely essay-driven,[2] computer science can benefit from a rich variety of assignment types, depending on what the course requires. Here are a few broad categories of assignment types. They are not exhaustive, and other disciplines will certainly have their own assignment types to add to this list. The purpose of this section is to illustrate how a variety of assignment types can facilitate each of the SIKE outcomes, and to show that the structure of supervision assignments are really completely up to the supervisor – it is entirely possible to experiment with new work formats with the aim of improving the quality of self-study.

- **Mathematical derivations and proofs**: these have the advantage of instantly being recognised by students as 'science-y'. Incorporating a few derivations or proofs for core concepts can provide good exam practice, as well as help students feel that they have a deeper appreciation for these core mathematical concepts.

- **Fact reproduction**: these ask questions of a factual nature. Answers to fact reproduction questions can span from a single word to several paragraphs. The key property of this type of questions is that they help students memorise core information.

  When you set and grade these questions, try and discourage students from copying definitions and facts verbatim from the lecture notes, if possible. This will not be possible if you ask students to reproduce a mathematical equation, for example, as I do in my supervisions for introductory artificial intelligence. The equation I ask students to state is Bayes' theorem. Naturally, the equation for Bayes' theorem they write is likely to be identical to the equation in the lecture notes.[3] So, in order to elicit a deeper engagement than merely copying, I ask that they also describe the core idea of Bayes' theorem *in plain English*, and contextualise it by stating an example that does not appear in the lecture notes. This, ideally, encourages students to memorise the equation primarily by what it *means*, and only secondarily by how it might be written down on paper.

[2] However, they don't have to be! If you supervise for a humanities course and wish to experiment with alternative forms of assignment, you should exercise the freedom that supervisors have to do so.

[3] If readers from other disciplines are confused why this is the case: the assigment is similar to asking students to state Einstein's equation of mass-energy equivalence; all students will write $E = mc^2$. There is really no point in expecting students to state it any differently.

- **Problem formulations**: problem formulation questions take an ill-defined real world problem and ask students to frame it in terms of a formal method they have been studying. This is, alas, emphasised far too little in the tripos, because these questions can have many different answers, and the answers are difficult to grade. Problem formulation questions are very much enjoyed by skills-motivated students. Moreover, these questions, if planned well, elicit a diversity of answers and make for good discussion during the supervision.

  Current and historical events, and commercial incentives in computer science and popular culture can make good targets for problem formulation questions. Consider the algorithms, techniques, proofs, methods that the students are learning. Can you think of real-world problems that have been or could be solved using those algorithms and techniques, where the connection between problem and solution is not straightforward? These can be presented to students as opportunities to test their ability to make connections. For example, in my supervisions for advanced artificial intelligence, I ask students how they would use reinforcement learning to train an intelligent agent to play the 2013 mobile game Flappy Bird.[4] The task is to take the complexity of the game and formalise it to a viable representation of the state space, the action space, and rewards.

  [4] https://en.wikipedia.org/wiki/Flappy_Bird

- **Further reading**: these ask students to read papers, articles, book chapters, or even videos, that you think are useful. They can be materials that better describe and reinforce concepts taught in the lectures, they can be additional extracurricular material, they can be case studies, they can be alternative perspectives to those presented in the lectures, or perhaps something you have written during the course of your research that you feel would be interesting to discuss with the students.

  Further reading is enjoyed by knowledge-motivated students and can help interest-motivated students as well. It is probably best not to assign very long readings (e.g., a paper or article longer than 10 pages). Readings should also be accompanied with a few written questions to help students read actively. If these are open-ended reflective questions (or perhaps if the material introduces alternatives to material taught in lectures, a compare-and-contrast style question would be appropriate), it is useful to let students know precisely how long you expect these answers to be in terms of words or pages.

- **Essays, long and short**: these are relatively uncommon in STEM disciplines and computer scientists are often uncomfortable writing long form answers. Nonetheless, this can be a valuable type of assignment to get students thinking critically about the subject. For instance, in my supervisions for introductory artificial intelligence, I set the essay question 'can computers think?'

These types of questions are valued by knowledge-motivated students, especially if you provide further reading as a backdrop for the essay prompt.

- **Programming**: having students actually implement and run algorithms and solutions can be very helpful, especially if the algorithm is presented purely mathematically in the lectures. The experience of translating from mathematical notation to code gives students a deeper, mechanical appreciation of the technique. Programming is valued by skills-motivated students. It also gives students something to bring to the supervision for a bit of show and tell. If you have allowed flexibility in the choice of programming language, then students can compare solutions and see how the same technique might be expressed/decomposed according to different programming paradigms.

- **Artefact generation**: programs are just one type of artefact that are of particular relevance to computer science. However, in other subjects, other artefacts might be of relevance. For instance, students might be asked to create literary artefacts (poems, short stories, etc.), or artworks (sketches, drawings, paintings, sculptures), or technical and design artefacts (diagrams, sketches, small scale models), etc.

- **Method practice**: a number of 'methods', i.e., techniques used in the professional practice of computer science and software design, are taught in the tripos. The best examples are found in the human-computer interaction course, which introduces methods such as cognitive walkthrough, low-fidelity prototyping, and controlled experiments.

  Having students conduct a cognitive walkthrough, for instance, can help ground a rather abstract idea in a concrete activity.[5] For supervisions, students can asked to conduct a cognitive walkthrough as part of self study, where the assignment is to conduct and write up the results of the walkthrough. Alternatively the homework component can be to *plan* the cognitive walkthrough (e.g., by describing the intended user and providing a rationale for which tasks might be informative to walk through), which is then conducted collaboratively during the supervision, with students taking notes and reflecting along the way. In the same way, other methods such as controlled experiments can be incorporated into supervision assignments. These are definitely valued by skills-motivated students.

[5] For those who have not encountered the cognitive walkthrough before: in brief, it is a method in which a computer system is evaluated (often by the developer of the system) by working through a series of tasks from the perspective of the intended user of the system, in order to understand whether the current design is understandable and easily usable. It is a cheap and effective way of evaluating websites, online portals, designs for kiosks, ATMs, and other simple interfaces.

Where can I get supervision assignments?

It can be quite challenging to develop a set of supervision assignments from scratch, but the good news is that you don't have to. There are resources available to inspire you and draw upon. Even though you must endeavour to ultimately create a supervision experience that is uniquely your own, it is helpful to see how others have solved this problem in the past. Here are some suggestions for how you might bootstrap your supervision assignments.

- **Tripos questions**: tripos questions are a *safe* option for new supervisors. Past tripos questions are easily accessible (at least for computer science), along with mark schemes from the lecturers. Tripos questions are easy to put a time estimate on – students are expected to spend 45 minutes on them during exams, and so it is reasonable to expect that they might take between 60 and 90 minutes to complete when done as supervision work. Tripos questions at least satisfy the requirement of helping prepare students for exams, and it can never be argued that tripos question practice is irrelevant or off-curriculum.

  It is quite possible to quickly assemble a 'cookie-cutter' set of supervision assignments from 3 or 4 tripos questions. While this might be a comforting format for new supervisors (if you base your supervision around a handful of tripos questions, you can't really go *wrong*), it is better for supervisors to develop a richer set of supervision assignments as soon as they gain confidence in their ability to supervise. Since tripos questions are designed for written exams, they are comprised almost exclusively of derivations/proofs and fact reproduction in computer science. Similarly, for many humanities subjects, tripos questions consist almost entirely of medium-length written essays. As outlined in the previous section, there is a rich diversity of assignment types that are often better suited to high quality self study.

- **Lecturers**: conscientious lecturers, at least for computer science, will distribute exercise sheets for their courses. These can either be obtained from the course webpage or by emailing the lecturer.

- **Other supervisors**: by scouring the web pages of the computer laboratory, it is possible to track down people who have supervised the same or similar courses in the past. Sometimes, these noble souls will post their supervision assignments on their websites for free use. Otherwise, requesting old work sets over email usually works. Does the lecturer for the course have any PhD students? Chances are, they will have been ~~press-ganged~~ gently persuaded into supervising the course and you can contact them for advice. Long-serving Directors of Studies will also be aware of previous supervisors. Student administration, particularly for

Part II courses, keeps rosters of previous and potential supervisors for each course. Drop them an email and ask if you can be connected to these people. For other disciplines: the method through which you track down other supervisors will vary, but the basic idea is the same – other supervisors have had to solve the problem of assigning work in the past, and it should help to gather information from them before you set about building your own assignments.

- **Reusing assignments**: if you have supervised the course in the past, your own previous assigned work sets are the perfect starting point. Try and ensure that they are kept up to date with the lecture course. Consider the lecture-supervision dynamic discussed earlier. Material is often delivered with different emphasis and in a different order each year. Along with my assignment plans, I keep notes each year about which assignments worked, which didn't, and what materials I would like to include in future years. These are a great resource when the time comes to revisit the assigned work and refresh them for a new year.

The fine balance of depth versus breadth

A key question you will find yourself revisiting repeatedly is how to balance breadth of coverage with depth. That is, should you aim for a set of assigned work that touches briefly on several aspects of the curriculum, or a set of assigned work that requires a deep, nuanced treatment of one or two specific aspects of the curriculum? There is no good way to answer this question. Breadth is better for exam-orientation, no doubt – if you detect that your students desire a thorough exam preparation, then they will prefer a 'minimum viable' treatment of the course, i.e., each aspect of the curriculum is touched upon with only as much depth as is likely to be expected in exam conditions. On the other hand, targeting depth is far better for skills and knowledge orientation.

Personally, I tend to lean towards a deeper exploration of a small number of concepts within the larger curriculum, with suggested optional work if the students would like the exam practice. This is based on my personal philosophy, that it is more important for students to carry with them a deeper appreciation of some concepts into their postgraduate lives, than exam preparation. This does not always align with what students want. I find this to be an acceptable compromise given that students are free to plan their own revision time in any case. Moreover, as outlined in the next chapter, there should still be ample opportunity in the supervisions to clarify any issues that the students are having with parts of the course not given the deep treatment.

Receiving work

Set a clear protocol for when, where, and how you would like to receive supervision work. My own guidelines are as follows, but they are just an example and will not apply to every discipline or even every computer science supervisor. Along with all the work for all supervisions, I send the following guidelines to all students at the very beginning of our correspondence:

1. Venue: supervisions will be at the Computer Laboratory (in the William Gates Building). We will meet in front of reception.

2. Deadline: work is due 48 hours before the supervision. Late work may not be marked. Excellent work submitted slightly late is preferable to timely but inadequate work. Work not handed in is immediately added to the work due for the subsequent supervision.

3. Submission:

   (a) Work is best handed in through email as a PDF attachment.

   (b) Do not send me links to hosted files.

   (c) Leave a margin of at least one inch on every side.

   (d) Any code or pseudocode must be in a monospaced typeface such as Courier or Monaco.

   (e) If your work is handwritten, a scanned copy (or legible pictures) is acceptable, but pay close attention to the next point.

   (f) Keep your total file size small (less than 0.5MB).

   (g) If you must hand in a hard copy, email me after you have put it in the blue box,[6] otherwise I won't pick it up.

   (h) The following must be clearly visible on the front page: your name, my name, the full name of the course, and the number of the supervision (e.g. 'Artificial Intelligence II Supervision 2 – John Doe for Advait Sarkar').

4. Support: if you have questions or difficulties related to the course, email me a few days in advance of the supervision with the query or difficulty to give me the best chance of addressing it well.

[6] This is a mailbox specifically for undergraduate supervision work used internally in the Computer Laboratory.

Each of these guidelines has been carefully considered, and sometimes has been introduced in response to specific events that actually happened. For instance, many supervisors ask for work only 24 hours in advance, and my 48 hour request might seem unusually harsh. However, firstly, this is much less of an issue for students if all the assignments are made known well in advance, and secondly, it allows me more time to mark them with care. Since I believe in

high quality self-study, I emphasise that meeting the deadline exactly is less important than doing excellent work. There is also a clear and simple instruction for students who do not hand in any work at all (not handing in work is discussed in more detail in the next chapter).

Moreover, I ask for:

- PDF submissions, since the appearance of the PDF I see is (almost) guaranteed to be identical to the one the student produced. This is not the case for Microsoft Word documents, for example, which can be mangled horribly when opened in a different version of Word from the one that authored it. All decent text editors / operating systems have PDF export built in, so this also gives students a greater freedom of choice.

- No links to hosted files (e.g., on Dropbox, OneDrive, and similar services), because links can break and change, and I don't want to have to mess about with downloading and keeping student files on my own filesystem – it is far better to quarantine them in mail folders.

- A margin, because there has to be room for me to write feedback. I don't ask for double line spacing because this is usually wasteful and actually not very pleasant to read.

Finally, I have found receiving queries and difficulties in advance of the supervision to be highly effective. This spares me the challenge of devising a thorough and satisfactory answer to a query on the spot in the supervision. Knowing the query in advance, I can spend time thinking about good illustrative examples, digging up alternative reading or video materials that explain the concept in a different way, preparing follow-up assignments to practice the concept, and just generally putting together a more comprehensive strategy to help the student conquer the difficulty decisively.

To reiterate, these are the guidelines that I have found useful for *my* supervisions, and different disciplines and supervisors will find that they prefer to give students different instructions. The key point here is that *you* are responsible for laying down these basic protocols and it is best done clearly, in writing, at the very outset.

Marking work

Always aim to return well-marked scripts to students, with detailed and individualised feedback on how they might improve their answers. The exception to this is if the student has handed in work far later than your deadline, in which case, don't feel obliged to mark it.

I try and frame my feedback as questions, such as 'What about ...? Have you considered the implications of ...? What would happen if instead ...?', so that addressing those questions would improve the answer. Students like to leaf through supervision work during supervisions, and so those questions can also become themes for discussion during the supervision, and provide good opportunities for peer teaching, as outlined in the next chapter. Be generous with praise, always be positive, and, with discretion, apply positive language such as 'nice', 'good', 'great', 'excellent', etc. Even if the student has clearly had a hard time with the work, seeing positive remarks about some aspect of the work will encourage them to participate in the supervision and has good implications for their general wellbeing and fragile sense of self worth.

It is a matter of personal taste whether you actually assign numeric scores to supervision work. There are some disadvantages: students may focus on the numbers rather than understanding concepts, students may compare marks with each other, and you will have to come up with an (essentially arbitrary) marking scheme for the assignments you design yourself. Other supervisors have developed simplified scoring systems (e.g., red/amber/green) and eschew totalling scores to mitigate some of these problems. Personally, I find it useful to assign numeric scores for two reasons. First, it forces me to take a reductionist lens to my own assignments, to plan ahead concretely how I would assess work handed in. Second, it helps me add quantitative information for each student to the supervision reports. Particularly low or high scores, along with descriptions of what went wrong or right, can give directors of studies and students a concrete basis for discussion and reflection.

5

A supervision

You've designed a smashing assignment. You've agreed on a time and place. Students have handed in their work (hopefully on time), and you have dutifully marked them and left helpful comments in the margins. Now all that remains is to meet for an hour. But what should happen in this hour?

The cookie-cutter supervision formula

Just as tripos questions make a 'safe' set of assignments for new supervisors, there is a 'safe' way to conduct supervisions that is difficult to go *wrong* with, but also difficult to achieve excellence with. This is as follows. Once your students have their work back in their hands, you systematically work through the assignment questions in order. You might skip trivial fact-reproduction tasks that no student had difficulty with, and spend most of your time on one or two particularly tricky problems (usually proofs or derivations) by writing on the whiteboard or having students write on the whiteboard. It is quite easy to spend an hour in this way. It satisfies the basic requirements of a supervision, but as with using tripos questions, should quickly be developed into a deeper and individualistic style of supervision. Individualised, that is, with respect to you, the supervisor.[1]

[1] Individualising supervisions with respect to students is, of course, a given.

The Socratic method

In my experience, I have found that my best supervisions all follow a similar pattern. The purpose of this section is to illustrate *one* alternative to the cookie-cutter formula that I have found to be particularly effective myself. It is not the purpose of this section to propose that this method is the best, or most universally applicable.

I have referred to this method using the term 'catechism' in the past, but I am now calling it 'Socratic method' because I wish to

avoid any of the religious connotations of the former. In practice, the method feels like a hybrid of catechism,[2] the literary technique of 'Socratic dialogue',[3] the dialectical Socratic method,[4] and the pedagogical method of 'Socratic questioning'.[5] I don't split hairs about the terminology because it has seemingly little effect on my practice, but if any philosophers or classicists would like to correctly define the method I shall now describe, please let me know.

The method is essentially to structure the supervision around a carefully designed sequence of questions that you pose to the students. The questions start as very basic, perhaps just restating fundamental facts, motivations, and axioms. Gradually, more facets are introduced to the basic questions, bringing in layers, counterexamples, and increasingly complex concepts. Some questions introduce major new ideas, and some merely flesh out ideas in greater depth, perhaps exploring counterexamples and edge cases.

As a toy example, let us imagine that the topic of the supervision is *addition*, that is, adding two quantities. This is just a simplistic example that I have chosen because it should cut across disciplines, not the real topic of any supervision I know to have been given.[6] A plausible exploration of the topic of 'addition' through the Socratic method can be done through the following questions:

1. Why do we want to add things?

2. What kinds of things can you add?

3. What kinds of things can't you add? Can you add a word to another word? Can you add a person to a person?

4. Can adding 2 apples to 3 apples be considered same thing as adding 2 oranges to 3 oranges? That is, does the nature of the objects you are adding affect the outcome? Why or why not?

5. How do we write down (represent) addition?

6. Does the order in which you add things matter?

7. What number, when added to any other number, leaves it unchanged?

8. What does it mean to add zero to a number?

9. What number can you add to another number to yield zero?

10. How does subtraction relate to addition?

11. What are negative numbers?

12. What is the limit to how much can be added to a number?

13. Which is the 'smallest' number: a negative infinity, or an infinitesimally small fraction just greater than zero?

14. How can we compute addition?

[2] https://en.wikipedia.org/wiki/Catechism

[3] https://en.wikipedia.org/wiki/Socratic_dialogue

[4] https://en.wikipedia.org/wiki/Socratic_method

[5] https://en.wikipedia.org/wiki/Socratic_questioning

[6] I had initially written: '[...] not the real topic of any Cambridge supervision ever given', but given that Cambridge has over 800 years of history, I probably would have been wrong. In fact, I would have almost certainly been wrong. Bertrand Russell and Alfred North Whitehead were Cambridge men. In their 1910-1913 *Principia Mathematica*, an enormously influential bid to describe axioms and logic from which all mathematical truths could be derived, they set forth several hundreds of pages of proof for the validity of the proposition $1 + 1 = 2$. I'm pretty sure at least some of their students had to sit through supervisions on the topic. Their broader endeavour to capture all of mathematics in a finite set of axioms was shown less than 20 years later, in 1931, by Kurt Gödel, to be impossible.

All this being said, my example is about 'addition' as taught at the middle school level, to 13-15 year olds – there probably haven't ever been any supervisions on that. Probably.

Notice how the questions come in groups. For instance, questions 7 and 8 deal with the concept of zero. Question 7 prompts students to come up with the concept of zero themselves, and question 8 probes further. Questions 9 to 11 deal with subtraction and negative numbers. Question 9 follows from question 7 in the sense that it has a suspiciously similar structure, and builds on the recently visited concept of zero, but introduces a completely new concept. Questions 10 and 11 probe further.

The supervision is then conducted by posing these questions, one after the other, to students, and allowing the discussion to unfold. In presenting these questions to students, and allowing them to discuss and explore answers, I find this gives them ownership of the solutions. There is no need to stick rigidly to the sequence; if the discussion moves from question 1 straight to question 4, allow it to happen, and revisit questions 2 and 3 at an appropriate moment.

It is important that the supervision assignment has laid the ground-work for these questions. Students should have already had to consider these questions in order to complete the assignment. However, they should not directly correspond to questions in the assignment, because then students can simply repeat the conclusions they have fleshed out in detail in the assignment. Rather, the answers to these questions should be implications, foundations, or corollar-ies of their assignment. Students who have done the work will be quick to realise the connection, and be able to discuss their answer with reference to the assignment. Moreover, this gives students who have faced difficulties completing an assignment the opportunity to revisit the assignment from a different perspective.

There's still room in Socratic dialogue for the hard whiteboard proofs and derivations (for example, in the toy example, the super-visor might wish to illustrate the concept of zero by working out what happens when you add zero to several different numbers), but here the sessions at the whiteboard are introduced as part of a discussion and a sequence of inquiries, rather than with the aim of merely exemplifying how a tripos question should be answered. This might seem obvious to those in some humanities disciplines, but at least for STEM disciplines, it is quite common for students to drown in the swamps of minutiae, amidst symbols, equations, and algorithms, without a sense of the bigger picture. This I know firsthand, as *I* was one of those students. For this reason, in assign-ments and during supervisions, I constantly insist that students discuss and rephrase core concepts using simple undecorated non-technical language as much as possible.

Paper is mightier than the whiteboard

I always sketch and write during supervisions, whether it is dia-
grams, charts, graphs, equations, derivations, pseudocode, or even
just a few scattered words to structure my thoughts. Many of these
sketches are done as part of the discussion, meant clearly as illus-
trative aids for the benefit of the students. I have experimented
with doing this both on the whiteboard and on loose sheets of
blank A4 paper. I have found that in most cases, using paper has
more advantages.

The advantages of the whiteboard are that it is easy to erase things,
it is easier to write larger, and the whiteboard is in the same orien-
tation for everyone.

The advantages of paper, on the other hand, are many:

- There are no body language 'power moves'. If you and your
  2-3 students are sitting around a table, you have to stand up
  and assume an authoritative position in order to write on the
  whiteboard. Similarly, if you ask a student to work through a
  derivation, they have to stand up in front of their peers, which
  can exacerbate the feeling of being put on the spot. With pa-
  per, you simply slide a sheet over to the student and they begin
  writing. This of course assumes that you are sitting together at
  a relatively compact table; it won't work if you're sitting around
  a wide rectangular table, where other people can't see the piece
  of paper in front of you. I find small circular or square tables,
  the kind designed for exactly four people, to be ideal not only
  for the logistics of a paper-driven supervision, but also because
  the proximity incentivises student engagement. It's much harder
  for a student to avoid participating when everyone is only inches
  away from everyone else.

- Besides the fact that one person standing while others sit cre-
  ates awkward authority structure, it can also waste time to re-
  configure the people in the room so that everyone is facing the
  whiteboard, especially if the room is small and chairs have to be
  shuffled around in order to yield access to the whiteboard. It is
  always smoother for the discussion to be able to seamlessly start
  writing and drawing without anyone having to move.

- You don't have to faff about with finding whiteboard markers
  that work, or whiteboard erasers. You can bring all the materials
  you need by yourself. I liked to carry one of those four-colour
  ballpoint pens, where one can quickly switch colours by pulling
  down one of four levers on the sides of the pen, so that I could
  quickly draw multi-coloured diagrams. This is possibly the only
  actual real-world use case for those ridiculous pens.

- You don't need a room with a whiteboard. This opens up a much wider range of venues for the supervision, such as the departmental café or outdoor benches – great for sunny springtime supervisions.

- At the end of the supervision, the papers become a resource that can be taken away by the students. With a whiteboard, the only viable option is to have students take pictures of the whiteboard as you go along, which they then have to remember to organise along with the rest of their notes.

Handling a lack of work

What do you do when students hand in very little, or no work at all? My protocol is straightforward, and is made known to students in my initial correspondence (see the previous chapter for advice on establishing supervision guidelines). If a student does not hand in any work, the work becomes due *in* addition to the work due for the next supervision. If multiple students are not ready with work, I might suggest that the supervision be rescheduled, but in this case quite often the suggestion to reschedule comes from the students themselves. I try not to intervene in scheduling decisions because it is hard enough to mutually agree upon a time in the first place.

I have seen supervisors employ many alternative strategies and protocols: they might contact the director of studies immediately, or they might cancel the supervision and take a more active role in rescheduling, or use a system of 'strikes' (e.g., no consequences for the first infraction, contacting the director of studies the second time, and so on).

Rarely, supervisors make discouraging remarks to the student believing falsely that this might spur them into better work habits. This is something that happened to me. It is completely inappropriate and most unwarranted. I struggled with supervision work throughout my undergraduate career, largely due to procrastination and lack of motivation, which trapped me in a negative feedback loop for much of my first two years. I was handing in work late, and incomplete, and for one particular supervision I just handed in no work at all. When I arrived at the supervision, in front of my two supervision partners, the supervisor looked at me directly and snapped angrily: 'next time, don't bother coming to the supervision'. I was, alas, too impressionable to realise that his behaviour was completely unacceptable. His attitude, rather than motivating me to work, caused me to believe that he had no interest in supervising me, and so I felt that there was nothing left to gain from his supervisions. *When students feel this way, the supervisor has failed*.

The important thing is to make no assumptions, and above all, to be empathetic. Vanishingly few students are deliberately lazy. For almost every case of missing or highly incomplete work, there is some issue that the student is working through, whether one of motivation, or physical and mental health, or simply a balance of priorities that meant that work for your supervision could not be done properly in time. You are not responsible for solving these issues. You are only responsible for supporting the student to develop their thinking and conduct high-quality self study. If a student appears to you to be having serious difficulty meeting deadlines, or has missed a supervision altogether without sending apologies, then discreetly notify the director of studies, describing the situation as non-judgmentally as possible. They can handle it from there on.

Handling imbalance of ability

You will inevitably have the experience of supervising a group of students who have varying levels of interest, work ethic, prior exposure to the subject, and articulacy. This will manifest as an imbalance of ability to complete the assignments and engage well in the supervision. I hate to express this simply as an 'imbalance of ability' because it implies that some students are inherently more able than others, which I do not believe, but I will use it as a convenient shorthand.

Handling an imbalance of ability is tricky, and there is no good general solution. There are at least two concrete problems that it poses: first, any explanations that you provide are necessarily compromised; a slow, in-depth explanation aimed towards one student might bore another, a fast overview that skips steps might leave one student behind, and an explanation somewhere in between benefits no-one. The second concrete problem is that one student will raise deep, knowledge-oriented questions that are inappropriate to discuss before the group as a whole first establishes some simpler groundwork concepts – and conversely, one student will require basic groundwork to be laid because it was not acquired during the lecture or self study. If there is a good rapport between supervision partners, peer teaching can be very effective here, as long as it is not overwhelmingly unidirectional, which would give the impression that the 'smarter' student is always being made to explain concepts to the 'less smart' student. Ensure that you elicit participation from all students equally.

Imbalance can also be addressed through tailored reading material and assignment options before and after supervisions. Strive to supply students who are experiencing difficulties with better basic materials, alternative lecture notes, illustrative videos, and references to other reading. Conversely, supply students who are

excelling with further reading, challenging mini-projects, and tantalising puzzles. This shifts some of the burden from you to split the difference between levels of ability during the actual supervision. Sometimes, offering to spend a few extra minutes with students having difficulties before or after the supervision to clear up outstanding doubts can also help.

As a case study that illustrates a number of interesting properties of imbalanced supervisions, let me tell you the true story of what I call the 'Goldilocks supervision', which concerns a supervision group of three students I had in my very first year of supervising. It was for a first year computer science course, essentially an introduction to programming, which is taken not only by computer scientists, but is also offered to first year natural scientists, as well as students taking psychology, politics and sociology. The Goldilocks group consisted of one computer scientist, one physicist, and one psychologist. The computer scientist excelled at assignments and supervisions, heading easily for a first; the natural scientist was average, heading for an upper second; the psychologist had serious difficulties with even basic concepts, heading for a third.

The first observation is that the difference in 'ability' was clearly due to the students' differing academic specialisations, rather than some innate cleverness. Computer scientists take several programming courses, theoretical and practical, and can easily relate material from this course to other things they are learning. In contrast, for psychologists, this course is a completely different paradigm where lectures are delivered in a different format, supervision work looks completely alien (writing disjoint snippets of computer code instead of a single essay), and nothing in the course content is relatable to anything else being learnt. Physicists fall somewhere in between.

At first during the Goldilocks supervisions, I found myself explaining concepts in triplicate. At any given moment I was speaking to only one student and effectively ignoring the other two. This wasted enormous amounts of time, and each student was basically bored for two thirds of the supervision. To tackle this challenge, I applied two of the strategies that I have outlined so far. The first was to spend small amounts of extra time before and after each supervision to assist the psychologist. The second was to curate separate lists of reading materials and exercises for each student to attempt in their own time. I had four supervisions with this group in Michaelmas. I did not see them again until Easter, when they requested a supervision for revision purposes. To my great pleasure, I found that the psychologist, having diligently worked through her recommended reading and exercises, was now significantly more comfortable with the material and headed towards a strong upper second.[7]

[7] This is just one of a number of my personal experiences that lend credence to David Mackay's assertion that 'Everyone Should Get an A', presented in a delightful short essay accessible here: http://www.inference.phy.cam.ac.uk/mackay/exams.pdf

Miscellaneous advice

- **Lead with student concerns**: when I begin a supervision, I always ask: "Is there anything that you found particularly difficult? Is there anything that we should definitely have discussed by the end of the supervision? Do you have any questions that you would like to have answered by the end of the supervision?" Be prepared to factor student answers into your supervision plan on the fly. Note that the questions are not phrased as an invitation to hijack your supervision plan, they are not for example "Did you have any questions that we should discuss now?", which would disrupt the narrative you have planned for the supervision. Rather, these opening questions are a reconnaissance activity where you gather input directly from your students to weave into your supervision plan, to ensure that any student needs you had not anticipated are indeed satisfied by the end of the supervision.

- **Quietness and timidity**: moderate the discussion so everyone has a chance to speak. Draw in students who are naturally reticent by addressing them by name, making eye contact, and asking a specific question. To avoid putting such students on the spot, try to use questions which are relatively straightforward, or ask for an opinion rather than a 'correct' answer. Don't overuse the technique of giving shy students straightforward questions, because that might come across as patronising, which will further embarrass the student.

- **Peer teaching**: as much as possible, have students explain concepts to each other. Before you launch into a discursive explanation, consider whether it is possible to ask one student to explain it to the rest of the group (it is almost always possible). When the student is done, recapitulate on their behalf. Balance the responsibility for peer teaching across your supervisees, ensuring everyone teaches and is taught.

- **Demographic factors**: be very sensitive to race, gender, socioeconomic status, neurological typicality, and other factors that might influence participation in a supervision. In your supervisions and your assignments, try to avoid picking unnecessarily inflammatory political, social, or religious examples.[8]

- **The lecture-supervision dynamic, revisited**: it might be useful to think of the lecture-supervision dynamic like a trip to a foreign country. Here, the lecturer is the tour guide, and you are an experienced local. The lecturer's aim is to give a broad overview of the topic, with necessary omissions and oversimplifications in order to present a coherent narrative. Your aim, on the other hand, is to show the students how the locals really live, and take them to interesting and niche places that they might have missed

[8] Easy to do in computer science, but perhaps not so easy to do in sociology or theology, but those people don't need me to tell them how to handle these sensitive issues anyway.

on the overview, but which enrich their awareness, ownership, and mastery of the topic.

- **Continuously assess understanding**: probe your students' understanding throughout the supervision. Ask short questions that can be quickly answered. Constantly ask if students are following, if they understand, if the concept is clear. You must also ensure that students actually understand when they say they do – they might sometimes nod along simply due to peer pressure, or because they *believe* they understand the concept but haven't thought through the implications. Establish this by asking follow up questions, or asking students to recapitulate in their own words.

- **Make notes before and after**: write at least three different types of notes. Before the supervision, write down notes about the work each student handed in. Was it submitted on time? Was it good? Was it complete? How much effort did the student appear to put in? What were clear problem areas and what were areas of excellence? After the supervision, write down notes about how each student participated in the supervision. Did they contribute to the discussion? How well did they answer the questions you asked? Did they learn well from peer teaching? Did they teach well when asked to do so? Finally, write notes about the assignments themselves. What worked? What didn't? Should you change the wording for next year? Should you place less emphasis on one aspect and more on another? Should you supply different reading materials next year?

Writing supervision reports

At the end of the supervisions, you are required to write concise supervision reports that will be seen by the student and their director of studies. At least three people benefit from supervision reports.

First, the student: the supervision report is an opportunity to celebrate the student's strengths, and summarise what you think the student's main weaknesses still are, and to recommend a course of action (e.g., through reading or assignments) that will help the student improve.

Second, the director of studies: the DoS is the only person who oversees the academic welfare of the student throughout their undergraduate career. Your reports should be written as useful representations of points in that career, so that the DoS can compare them with reports from other supervisors, to reconstruct the students' trajectory. I always mention what the quality of the work was (with marks if appropriate), whether it was handed in on time, how

the student behaved during the supervision, how they responded to feedback, and any highlights or low points in our interaction.

Third, and finally, yourself: the process of writing the report is an opportunity for you to reflect and assess how well you have served the needs of the student (although you won't be assessing yourself explicitly in writing, of course).

Writing supervision reports is often regarded as a chore, but if you have been diligently writing notes before and after supervisions, as recommended, then you should already have plenty of raw material, which should require only light redrafting, to write your supervision reports.

6

The project

This document has been primarily about supervising *courses*. However, in several disciplines, there is also another type of supervision which is a completely different experience. It is time consuming, and a year-long commitment, but can be ultimately deeply fulfilling for both student and supervisor. I am talking, of course, about supervising final year projects. This brief chapter presents a few thoughts about supervising projects.

What is the purpose of a project?

Final year undergraduate projects typically contribute a large portion (anywhere between 15% and 30%) of the students' final grade. They are, sadly, often the only opportunity a student gets to influence their final grade outside of the exam setting. As such, a trivial answer to 'what is the purpose of a project' can be simply 'to give students an opportunity to get as high a grade as possible outside the exams'.

However, the student will spend an entire year on this project, as, in effect, will you. So it is very worthwhile considering what the student wishes to get out of the project, and to ensure that it is a sensible goal, and that you will be able to assist them in achieving that.

CV BUILDING: sometimes, the student is interested in having a flagship item in their portfolio that directly assists with their job search. For instance, students wishing to get jobs in data science or machine learning may want to work on a project that involves popular contemporary machine learning technologies, not only to boost their expertise, but also to have some evidence of that expertise on their CV. The project should be designed to correspond well to a known problem with market value. In this case, you should help the student use their work to build a strong addition to their

portfolio – perhaps encourage them to develop their code in a modular fashion so that it can be open-sourced on GitHub. In some sense, the gold standard of a successful CV building project in computer science is software that becomes widely used by other people, perhaps as an open-source project, or as a commercial venture. In other disciplines, you might encourage them to discuss and present their work at college events, student conferences, etc., or perhaps create a webpage or video summarising the work.

Research experience: Sometimes, the student is interested in getting research experience, most likely because they intend to apply for Master's or PhD programmes. In this case, rather than market value, the project should be about showcasing academic novelty and potential for contributions to knowledge. Undergraduates are unable to formulate viable research programs for themselves so you will have to assist them here. You might consider giving them sub-projects from your own research agenda that could plausibly be completed by an undergraduate. The gold standard of evidence for a 'good' research-focused final year project is, of course, a paper in a good conference or journal. This is the standard I expect of all of my students, which might seem a bit much for undergraduates, but my students, I am very proud to say, have repeatedly matched this expectation. Some have even won awards for their work, and gone directly into PhD programmes, research assistantships, or industrial research.

Let me also just take this opportunity to state emphatically that the student should *always* be given first authorship on any papers arising directly from their work. If they have joined a large project of which their work formed only a small part, then by all means include them as a middle author on the 'big' paper reporting the entirety of the project – but another paper should be written focused more directly their work, with them as first author. If there is not enough material from their project for this, you have failed to scope the project ambitiously enough. I had better stop this paragraph now before it turns into a rant on my philosophy on authorship.[1]

Deeper exploration: Sometimes, the student wants neither CV building nor research experience, but is simply interested in understanding a subject more deeply. In this case, you must work together to fashion a project that involves wide reading and exploration, while still falling within the parameters of the project specification.

One strategy I have found useful in the past is to build a project based on the student's previous experience. Perhaps they had an interesting internship, or conducted an interesting independent study. Building upon this work to extend it can be very effective, since some work has already been done, the student is familiar with the tools and techniques required, and they might already have

[1] Don't even get me started on mathematicians' use of alphabetical ordering, massively collaborative papers with hundreds of authors, or the bizarre imaginary concept of 'joint first authorship'. My current ideological stance on authorship is that every author must have made a *significant contribution* to the *marginal intellectual effort* of producing the paper, with order strictly dictated by level of contribution.

the relevant contacts and access to information. All that remains is for them to simply invest the additional effort required to answer the specific new research question or solve the newly formulated project objective. Is this 'cheating'? No. It is an effective use of prior expertise and resources. The student is still doing substantial, novel work. However, they avoid the startup costs of organising the resources and logistics, and learning how to use tools and techniques. This allows them to spend more time doing actual work, and consequently achieve much more than would normally be expected of students in this short time. In fact, I think all final year undergraduates would benefit greatly from starting to explore their project topic through work experience or independent study in the previous summer.

Meetings

How often should you meet with the student, and what should be the subject of the meeting? This varies enormously by discipline. Some disciplines have strict regulations about how often students and supervisors should meet. In the extreme case, some undergraduates are only allowed to meet their supervisors twice – in the entire year! This strikes me as barbaric, but that's what they've decided upon for themselves, so hey-ho. The computer laboratory does not have any strict rules. When I was an undergraduate, I met with my supervisor for around 10 minutes fortnightly. I thought this was far too little, so now I meet with my students weekly, for at least half an hour, up to an hour.

During the meeting, we discuss how the work is progressing, and if there are any hurdles the student is experiencing. We discuss potential solutions to hurdles, and agree upon a plausible plan of action. We discuss how the project is going in relation to the overall timeline of the year, and what we might do to help a project that is running behind, or what we might be able to do with the extra time when a project is running ahead (it happens!). As much as possible, I reinforce that the student is doing a good job, and that the project has the potential for excellence. Students leave the meeting with a clear set of priorities for the next week that they have set themselves. Then, in following week's supervision, we reflect on the priorities set the previous week and whether the student was able to achieve those. There will be weeks where the student has not made any progress at all on the project – this is to be expected, and handled on a case by case basis. If you have reason to believe that the student is in risk of not completing the project, then work together with the student and their director of studies to devise a reduced scope for the project and a concrete set of suggested further steps. If the student is confident that they will be able to make up for lost time, then they should be given the

benefit of the doubt.

Is this too much handholding? No. This is what it takes to achieve excellence. I *never* direct a student project on their behalf, or freely offer solutions and answers – it is their job to discover solutions. I am there to highlight relevant literature, expand their thinking, and help them keep their project on track. I am also there to show them that we at the computer laboratory take undergraduate education very, very seriously, and we believe that their project is important and worth doing. Expecting undergraduates to work in isolation for the entire year, receiving only occasional bits of interaction and feedback from an ivory tower don, seems like a particularly good way to tell them that their work is not valued. The effect of this divergence of approaches is plain to see – our computer science undergraduates publish their work at influential venues, are the most employed in the country, and win awards. Not so, unfortunately, for many other disciplines.

Time management and writing

Helping students manage their time can make a huge difference to their experience of doing the project, not to mention the quality of the final project. Dissertations, at least for computer science, are submitted in Easter term. However, students should have ideally finished writing the first draft of the dissertation over the Easter holidays, so that they only have to spend minimal effort on their dissertation during Easter term itself. This takes a lot of pressure off of them and allows them to focus on the all-important final year exam preparation.

Have students work backwards from this first draft rule-of-thumb to generate plausible milestones for what they can achieve in Michaelmas, the winter break, and Lent terms, during which they should have completed the research/engineering work required. This is somewhat of a STEM-centric view, where the act of research can meaningfully be separated from the act of writing the dissertation, but the general principle is to aim for a completed first draft before Easter term begins, and work backwards to formulate milestones. Computer science students already do this as part of their initial project proposal.

7

Conclusion

Supervisions are the greatest pedagogic treasure of the Universities of Cambridge and Oxford. The fact that the supervision system works at all is a logistical marvel. All supervisors are, in a small way, curators of this system – not only are they directly responsible for the undergraduates seated in front of them, but they are also responsible for all who come afterwards, as their words and deeds shape the meaning of what it is to supervise and to be supervised.

This brings us to the end of my reflections on supervising. My aim has been to be descriptivist, as a product and now curator of 'the' supervision as a lived experience for nearly eight years. I hope this has been a helpful and insightful resource. Even as you read this, thousands of supervisors and students across Cambridge are grappling, whether they know it or not, with the questions of what a supervision even is, and how they can make the most of it. Everyone must answer this question for themselves, and I hope that I have at the very least engendered a critical appreciation of the issues surrounding supervisions, and shifted your perspective from supervisions as being things that happen to you, to supervisions as being things that you make happen.

I wish you the best of luck in your endeavours.